Visual and Inertial Data Fusion for Globally Consistent Point Cloud Registration

Cláudio dos Santos Fernandes Erickson Rangel do Nascimento Mario Fernando Montenegro Campos
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
E-mail: {csantos,erickson,mario}@dcc.ufmg.br

Fig. 1. Walls of a 9.84m × 7.13m room reconstructed using the proposed methodology. The figure shows the keyframes of the globally optimized map with each point represented by a surfel.

***Abstract*—This work addresses the problem of aligning a set of point clouds acquired by an RGB-D sensor. To achieve this, we propose the fusion of RGB-D data with the orientation provided by a MARG (Magnetic, Angular Rate and Gravity) sensor, a combination that hasn't been extensively explored in the literature.**

Our methodology uses MARG data both in the coarse pairwise alignment between point clouds and in the loop closure detection between key frames. In our experiments, we were able to align the walls of a room with dimensions 9.84m × 7.13m. Our analysis shows us that MARG data helps to improve the alignment quality.

***Keywords*-Point Cloud Registration; Inertial Sensing; SLAM**

I. Introduction

The general public has recently witnessed the popularization of low cost 3D scanning devices, with prices that range from US$100.00, for an Xbox Kinect, to US$2,995.00 for a NextEngine 3D scanner [1] – and, despite their low prices when compared to industrial scanners, these devices operate with relatively high accuracies, with errors that are typically as low as 0.0127cm for the most expensive models.

The introduction of those fast and inexpensive 3D scanning devices has brought about a wide range of applications in several areas, from entertainment and sports that require accurate range measurements, such as golf and archery to digital 3D modeling. The entertainment industry has a growing requirement for digitally modeled objects and characters, which, in some circumstances, is achieved by acquiring 3D scans of sculptures and even living actors. There are companies specialized in digitalizing sculptures and then reproducing faithful replicas of the original piece of art.

Another application that benefits from 3D scanning is the monitoring and the following up of construction sites in order to verify compliance with CAD models. Point clouds obtained from those environments can be submitted to a cross validation procedure, also known *as built*, that is capable to detect relevant differences between the CAD project and what has been actually built [1]. Conversely, this technique can be used, for instance, to recover CAD models from historic buildings.

Inertial measurement units (IMU) – devices that are able to measure attitude (angular orientation in space), are becoming increasingly popular and inexpensive. Since the release of the Wii console, in late 2006, inertial sensors became very popular and have also been integrated to smart phones, MP3 players and other console peripherals. Thus, thanks to the adoption of these sensors in different commodity devices, great research effort has been devoted towards improving their accuracy and lowering their cost.

Therefore, one can expect these sensors to serve complimentary purposes in a mapping framework, as the problem of environment mapping can be broken down into grabbing several local depth images and assembling them into a global representation. The later requires an estimate of the pose of the depth sensor at the time the depth images were acquired.

Our goal with this work was to develop and evaluate a methodology for fusing data from a sensor that captures both color and geometry (RGB-D sensor) and a variant of inertial sensors called MARG (Magnetic, Angular Rate, and Gravity) in order to produce a globally consistent environment map. The approached problem is relevant to the computer vision field,

[1]Market prices as of February, 2013.

since it has impacts on several areas that perform 3D modelling by using scans obtained from depth sensors. Amongst these areas, one can mention the digital replication of sculptures and art objects, the modelling of characters for games and movies, and even the reconstruction of CAD models from old buildings. It is also closely related to robotics, since its solution may allow mobile robots to map, localize and navigate autonomously in unknown environments.

Our approach extends a photoconsistent alignment methodology by introducing a coarse alignment stage that uses depth, color and attitude data, increasing its robustness to color and geometric ambiguities. We also propose a loop closure detector that takes advantage of the attitude data available, with which we can perform a globally consistent map optimization.

A. Related work

The problem of point cloud alignment impacts several fields, being particularly relevant to computer vision and robotics. We focus on one of its most common applications – environment mapping – in which the objective is to capture a representation of an environment using a sensor (for instance, a 3D scanner or a sequence of images). It is important to note that not all solutions to the environment mapping problem rely explicitly on point clouds alignment, and they depend largely on the type of available sensors.

Both in computer vision and in the mobile robotics fields, several researchers have been studying the Simultaneous Localization and Mapping (SLAM) problem. There are several SLAM approaches in the literature, but it can be simply stated as to incrementally build a map of the environment, while obtaining the pose of the sensor with respect to the environment. In mobile robotics, one important task is to build a map where the robot can localize itself. Depending on the task, environment maps can be either two or three dimensional. Simple structured environments, such as a room or a single floor of a building might be easily represented by a 2D map, while more complex environments, such as mines and multistory buildings, may require a 3D representation.

There are several possibilities sensors that may be used for the mapping. Some researchers, for instance, have successfully used color cameras to build maps of static scenes, due to their typical large field of view and low cost. Classic vision-based mapping systems seek to build a 3D map based on features detected in the images captured from a camera moving in the environment [2]. In particular, [3] performed real-time, drift free, visual SLAM with high quality features, but limited to a small amount of features, which would be practical only for mapping small environments. Other methodologies with lighter spatial restrictions have already been used to generate a voxelized representation of 3D features for the purposes of robot navigation, map visualization, etc. [4].

Some methodologies have mitigated the computational problem by running the mapping process at lower rates than the localization step [5], while the scalability issue has being addressed by separating the environment into several keyframes, which can be separately processed [6].

In general, vision-based mapping techniques provide good localization estimates with a relatively low computational cost when sparse features are used. Despite not being sophisticated enough to provide detailed environment maps, this technique can be already used for real time accurate localization in small to mid scale environments. The advances made in this field throughout the years also present great value to approaches based on multiple sensors, such as the one we proposed in this paper.

We address the environmental mapping problem as an instance of the globally consistent registration problem. The simplest version of this problem, in which only local consistency is sought, has been approached in many different ways in the literature: By using the Principal Component Analysis to find the relative rotation between point clouds [7], by discretizing the space and trying to match statistically similar voxels [8], [9] or by matching only three control points [10], to name a few. One of the most popular registration algorithm is the Iterative Closest Points proposed by [11]. During each iteration, this algorithm finds the correspondences between the point clouds and computes the affine transformation that minimizes the RMS error of the distances between pairs points in the clouds. Several registration techniques only use geometric information (i.e 2D or 3D point clouds), although recent researches have sought to use color information in this process due to the popularization of RGB-D sensors.

There are several recent methodologies developed for RGB-D sensors (such as the Kinect and XTion). They are typically focused on fusing color information with depth data to perform pairwise alignment of frames that can be later optimized for global consistency [12], [13], [14]. In spite of being driven by the popularization of this sensor, registration algorithms that use color information are not new, and significant results have been published years before these sensors were released [15]. However, it was only recently that, enabled by the computational power provided by GPUs, real time RGB-D mapping has been made possible. The KinectFusion [16], for instance, accomplishes this goal but it is limited to small volumes, which has been further improved by [17]. Their methodology allows the mapping of scenes of arbitrary sizes by generating a polygonal mesh of regions that fall outside the central volume.

Although the use of visual information together with depth data helps to increase the quality of 3D alignment, mapping systems based solely on visual features have their kinematics significantly constrained, since it can be very difficult to recover from divergences which may occur when the sensor moves too quickly. Furthermore, the number of correspondences between consecutive frames tends to decrease as motion velocities increase. Also, visual features may be lacking under certain lighting conditions, which is a key issue for applications such as prospection of uninhabited environments.

In order to overcome these problems, some methodologies approach the mapping problem by combining inertial data with 3D information from laser scanners [18], [19]. Although those scanners are able to acquire data from larger volumes than a

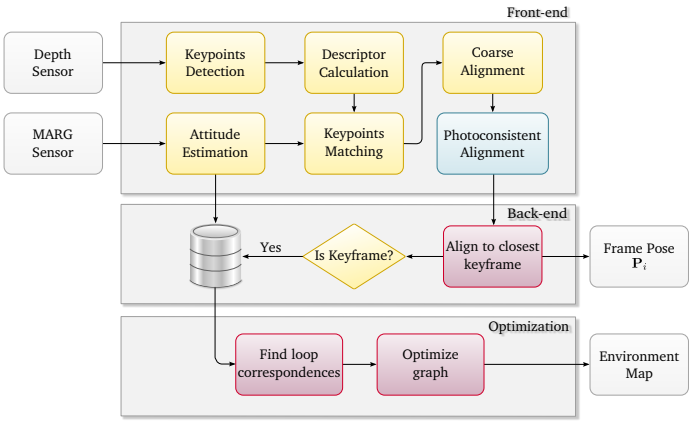


Fig. 2. High-level overview of the proposed approach. The front-end is divided into three pairwise registration levels, and their result is refined by a SLAM optimizer.

RGB-D sensor, they cannot capture color from the environment, and have a significantly higher cost.

To the best of our knowledge, the work presented by [20] is the only one reported in the literature to integrate RGB-D and inertial data to tackle the environmental mapping problem. Inspired by the work [21], it introduces the data fusion with the inertial pose estimated after a coarse alignment, which is performed by matching visual features from different sensor frames. However, [20] depends on the difference between the estimated attitudes from both the IMU and the key point matching process, discarding IMU data if the difference is too large. This means that only closely redundant information are fused, rendering the IMU data useless in many situations. Moreover, they do not address the loop closure problem, which we we have developed in ours.

II. Methodology

In this section, we describe our approach in detail. We propose a registration technique that consists of a coarse-to-fine alignment pipeline. This design has low computational requirement, since transformations coarsely computed by initial stages reduce the number of iterations that may be required to refine to the optimal solutions. Figure 2 depicts a high level view of our approach.

The first step in our methodology is the key points detection and descriptor calculation for each point cloud captured. Then, the whole pipeline is executed, whilst the alignment processes is performed between the last two two point clouds in a pairwise fashion. Each alignment level will be described in detail in the following subsections.

It is assumed that the RGB-D and IMU sensors were correctly calibrated in a previous step, and that their data is synchronized.

A. Coarse Alignment

In the first alignment stage, our methodology computes an initial estimate for the rigid transformation between the two latest point clouds by using inertial, texture and geometric information.

In order to accomplish the fusion of texture and geometric information, we firstly detect a set of SURF keypoints on the most recent image, and label them with BASE descriptors [22]. This descriptor is computed by using both texture and geometry information, which makes it more robust to variations in lighting conditions and textureless scenes.

We then match the keypoints from the two last point clouds. We developed a fast keypoint matching strategy that takes into account the *a priori* knowledge of the displacement between these point clouds. This prior knowledge is extracted from estimation of the rotation between two point clouds given by the MARG sensor, and the translation that can be roughly estimated by finding any pair of corresponding keypoints.

Let ${}^{k}_{m}\mathbf{R}$ be the MARG rotation w.r.t. the depth sensor (obtained from an extrinsic calibration step), and ${}^{w}_{m}\mathbf{M}_t$ be the MARG attitude w.r.t. a world fixed frame when the t-th point cloud was captured. If the MARG sensor was not susceptible to noise and interference, the attitude of this point cloud w.r.t. the $(t-1)$-th point cloud would be given by

$$ {}^{k_{t-1}}_{k_t}\mathbf{K} = ({}^{w}_{m}\mathbf{M}_{t-1}\ {}^{k}_{m}\mathbf{R}^{-1})^{-1}\ {}^{w}_{m}\mathbf{M}_t\ {}^{k}_{m}\mathbf{R}^{-1}. \quad (1) $$

Here, m represents the MARG frame, w the fixed world frame and k the depth sensor frame. In practice, Equation 1 doesn't yield an exact attitude due to problems such as noise from the MARG sensor, small extrinsic calibration errors and the sync variations between MARG and RGB-D sensor. Furthermore, it only accounts for rotation, leaving the translation unknown. We tackle these issues by combining the estimated attitude with a maximum sensor motion speed assumption. Given a keypoint ${}^{k_{t-1}}\mathbf{k}$ on the $(t-1)$-th frame, its corresponding match on frame t is likely to lie within a spherical shell defined by $({}^{k_{t-1}}_{k_t}\mathbf{K})^{-1}\ {}^{k_{t-1}}\mathbf{k} + \mathbf{r}v_{\max}dt$, where $\mathbf{r}$ is a unit vector, $v_{\max}$ is the maximum motion velocity assumed, and dt is the elapsed time separating the acquisition of those frames.

Considering the maximum velocity assumption, we select a subset composed of the N most prominent keypoints from the $(t-1)$-th frame (keypoints are compared by their response to the SURF detector). Then, for each keypoint, we search for correspondences on the t-th image. To compute the set of matching candidates from the t-th frame, we perform a radial search for key points around a center given by $({}^{k_{t-1}}_{k_t}\mathbf{K})^{-1}\ {}^{k_{t-1}}\mathbf{k}$ and radius of length $v_{\max}dt$. Each candidate in this set has an associated vector $\mathbf{t}$, which represents the translation between the two frames.

We proceed by comparing all combinations between an arbitrary ${}^{k_{t-1}}\mathbf{k}_i$ and its corresponding results from the radial search. For each pairing, we have a different $\mathbf{t}$ that, together with the attitude guess from the MARG sensor, provide an estimate of the pose of point cloud t w.r.t. point cloud $t-1$. Then, in order to refine this estimate, for all remaining key points ${}^{k_{t-1}}\mathbf{k}_l$, we perform a nearest neighbor search in the t-th frame around the point given by $({}^{k_{t-1}}_{k_t}\mathbf{K})^{-1}\ {}^{k_{t-1}}\mathbf{k}_l + \mathbf{t}$ – we

have found that three neighbors is typically a good compromise between computational performance and alignment quality. The neighbor that best matches the current keypoint is then defined as its pair. Once we have a correspondence for every keypoint subsampled from the $(t-1)$-th frame, we calculate the *Hamming distance* between the binary descriptors of the matched key points. The distances are used to calculate a score of the current pairing configuration. We repeat this process for all possible ${}^{k_{t-1}}\mathbf{k}_i$, and chose the pairing configuration with the smallest score. Algorithm 1 depicts the whole procedure.

Since the quality of a pairing configuration is proportional to the number of matches found (a false-positive is less likely to have a large number of corresponding key points) and inversely proportional to the Hamming distances of their descriptors, we defined our score to be the average of the Hamming distances divided by the number of matched key points.

Algorithm 1 Key point Matching

Require:

1: Keypoint sets ${}^{k_{t-1}}\mathbf{k} = \{{}^{k_{t-1}}\mathbf{k}_1, \ldots, {}^{k_{t-1}}\mathbf{k}_n\}$ and ${}^{k_t}\mathbf{k} = \{{}^{k_t}\mathbf{k}_1, \ldots, {}^{k_t}\mathbf{k}_m\}$ with their corresponding descriptors.
2: The attitude of the depth sensor at the instant t w.r.t instant $t-1$, ${}^{k_{t-1}}_{k_t}\mathbf{K}$.

1: ${}^{k_{t-1}}\hat{\mathbf{k}} \leftarrow$ SUBSAMPLE(${}^{k_{t-1}}\mathbf{k}$)
2: bestScore $\leftarrow \infty$
3: **for all** ${}^{k_{t-1}}\hat{\mathbf{k}}_l \in {}^{k_{t-1}}\hat{\mathbf{k}}$ **do**
4: ${}^{k_t}\hat{\mathbf{k}} \leftarrow$ RADIUSSEARCH(${}^{k_t}\mathbf{k}$, $({}^{k_{t-1}}_{k_t}\mathbf{K})^{-1}\ {}^{k_{t-1}}\hat{\mathbf{k}}_l$, $\forall \cdot dt$)
5: **for all** ${}^{k_t}\hat{\mathbf{k}}_l \in {}^{k_t}\hat{\mathbf{k}}$ **do**
6: score $\leftarrow$ HAMMING(DESC(${}^{k_{t-1}}\hat{\mathbf{k}}_l$), DESC(${}^{k_t}\hat{\mathbf{k}}_l$))
7: matches $\leftarrow \{({}^{k_{t-1}}\hat{\mathbf{k}}_l, {}^{k_t}\hat{\mathbf{k}}_l)\}$
8: $\mathbf{t} \leftarrow {}^{k_t}\hat{\mathbf{k}}_l - ({}^{k_{t-1}}_{k_t}\mathbf{K})^{-1}\ {}^{k_{t-1}}\hat{\mathbf{k}}_l$
9: **for all** ${}^{k_{t-1}}\hat{\mathbf{k}}_m \in {}^{k_{t-1}}\hat{\mathbf{k}} \mid m \neq l$ **do**
10: $\mathbf{c} \leftarrow ({}^{k_{t-1}}_{k_t}\mathbf{K})^{-1}\ {}^{k_{t-1}}\hat{\mathbf{k}}_m + \mathbf{t}$
11: neighbors $\leftarrow$KNN(${}^{k_t}\mathbf{k}$, $\mathbf{c}$, HARD_LIMIT)
12: ${}^{k_t}\hat{\mathbf{k}}_m \leftarrow$ BESTNEIGHBOR(neighbors)
13: hamming $\leftarrow$ HAMMING(DESC(${}^{k_{t-1}}\hat{\mathbf{k}}_m$), DESC(${}^{k_t}\hat{\mathbf{k}}_m$))
14: **if** $\|\mathbf{c} - {}^{k_t}\hat{\mathbf{k}}_m\| \leq$ THRESHOLD **then**
15: score $\leftarrow$ score + hamming
16: matches $\leftarrow$ matches $\cup \{({}^{k_{t-1}}\hat{\mathbf{k}}_m, {}^{k_t}\hat{\mathbf{k}}_m)\}$
17: score $\leftarrow$ score/SIZE(matches)^2
18: **if** bestScore $>$ score **then**
19: bestScore $\leftarrow$ score
20: ${}^{k_{t-1}}_{k_t}\mathbf{P} \leftarrow$ POSEFROMPCA(matches)
21: **return** $({}^{k_{t-1}}_{k_t}\mathbf{P}, \text{bestScore})$

To keep the computational cost of search operations low, the set of keypoints from point cloud t is stored in a KD-tree, allowing for quick range and nearest neighbor searches.

Differently from brute force techniques, we designed our matcher to apply a transformation to the keypoint around which a neighborhood search will be performed such that, after the transformation, the keypoint will lie closer to its correspondence. Also, the distance threshold used after the KNN search can be set to tight values in order to preserve euclidean constraints, thus eliminating the necessity for an explicit outlier removal step.

B. Photo Consistent Alignment

The acquired frames often contain a substantial amount of color information, which can be densely used to further improve the coarse alignment, correcting for small displacements on the pair-wise registration.

In our methodology, we refine the coarse alignment by using the approach described by [13]. Given two point clouds ${}^a\mathbf{C}_a, {}^b\mathbf{C}_b$ where each point contains its color, and an initial guess for the pose ${}^b_a\hat{\mathbf{P}}$ of a with respect to b, this approach searches for the pose ${}^b_a\mathbf{P}$ that leads to the smallest difference between the image from ${}^b\mathbf{C}_b$ and the image obtained after applying the perspective projection to the points ${}^b_a\mathbf{P}{}^a\mathbf{C}_a$.

In our pipeline, this alignment stage outputs a refined pose matrix, ${}^{k_{t-1}}_{k_t}\mathbf{P}$, which is accumulated into the global sensor pose ${}^{k_1}_{k_t}\mathbf{P}$ by the product ${}^{k_1}_{k_t}\mathbf{P} = {}^{k_1}_{k_{t-1}}\mathbf{P}\ {}^{k_{t-1}}_{k_t}\mathbf{P}$. The accumulated sensor pose is then passed on to the back-end processing stage.

C. Sampling Keyframes and Graph Optimization

The major purpose of our back-end is to select a subset of the captured point clouds, which we refer to as *keyframes*, and build a globally consistent representation of the environment with them. This is necessary because pairwise registration of point clouds tends to accumulate errors that greatly worsens the reconstruction result as the mapped region increases.

To overcome this issue, our back-end is capable of detecting overlapping regions between keyframes, especially when the sensor returns to a previously mapped region, which may be very difficult to detect since the uncertainty due to the accumulated error can make the search space too large when matching temporally distant keyframes. Additionally, our back-end tries to prevent regular point clouds from diverging indiscriminately from its closest key frame, which is accomplished by aligning each point cloud to the last detected key frame. The two algorithms that accomplish this are described in detail below.

1) Key frame Detection: The criteria for determining whether a point cloud can be considered a key frame or not is crucial for any registration system. On the one hand, a large amount of keyframes may cluttering the graph to be optimized, leading to higher running times and increased accumulated error after mapping the whole environment. On the other hand, a small number of keyframes conduce to a smaller intersection region between them, which turns the alignment into a more difficult task, which increases the chances for registration divergence.

With that in mind, our key frame detection policy takes into account the area of the region of intersection between a candidate point cloud and the rest of the keyframes. This metric allows us to address the divergence issue, since a good registration depends on the size of the overlapping region, but also gives control over the amount of detected keyframes, which can be reduced by decreasing the intersection threshold.

2) Alignment to Closest Key Frame: The alignment error between a regular point cloud and its closest key frame is typically small enough that both the estimated translation and rotation can be used as an initial guess for the frame-to-key-frame alignment, which is accomplished by a slightly modified version of the key point matcher algorithm previously described. Given a pose estimate of the t-th frame with respect to the key frame f, ${}^{f}_{k_t}\hat{\mathbf{P}} = {}^{k_1}_{f}\mathbf{P}^{-1}\ {}^{k_1}_{k_t}\mathbf{P}$, the radial search of the key point matcher is performed around ${}^{f}_{k_t}\hat{\mathbf{P}}^{-1}\ {}^{f}\mathbf{k}$, ${}^{f}\mathbf{k} \in f$.

D. Optimizing Environment Graph and Closing Loops

In the optimization step, we associate a hyper graph to the key frame structure, where each key frame is represented by a vertex, and each key point correspondence between frames becomes an edge between their corresponding vertices.

An important aspect of a global alignment procedure is the detection of loop closures between two candidate keyframes. Our approach uses the attitude reported by the MARG sensor, since it does not suffer from significant drift under the assumed operational conditions. Therefore, our methodology reduces the loop detection problem to finding a translation that connects two candidate keyframes, while being able to reject candidates with a large angular displacement. This is done with an algorithm that follows the idea behind our key point matcher, that is, the transform between keyframes is estimated from a set of corresponding key points.

The keypoint correspondences are computed using another modified version of the keypoint matcher employed in the coarse alignment. Given two keyframes i and j, our matcher is divided into two steps. Firstly, all key points from a subset ${}^{k_i}\hat{\mathbf{k}}$ are tested against all keypoints from a subset ${}^{k_j}\hat{\mathbf{k}}$. These subsets are obtained by dividing both keyframes i and j into subregions and extracting only a limited amount of keypoints from each subregion. During each single test, we estimate the translation between i and j by using the chosen keypoints coordinates and the relative MARG attitude between these keyframes. Similar to the coarse keypoint matcher, this transformation is used to match all remaining key points, and a score is computed for the current transform using the same metric used before. The estimated transform is stored in a priority queue in which the weight is defined by the alignment score. Finally, in the second step, all transforms in the priority queue with a score below an arbitrary threshold are submitted to a matching strategy where all keypoints from both keyframes are taken into account, instead of the subsampled sets.

Since the hyper graph is locally consistent, the edges between neighboring vertices can be those provided by the pairwise alignment stage. As far as the non-neighbor vertices are concerned, we use orientation data from the MARG sensor to discard matching candidates that fall below a given threshold of the angle between their view direction vector. For instance, a loop cannot be closed between two keyframes if the first one was captured with the sensor pointing towards the floor and, the second, pointed to the ceiling.

III. Experiments

Our experimental analysis seeks to compute the fidelity of a map generated by the proposed methodology. Our ground truth was captured by a Zebedee [19] device. We are also concerned with how each component of our system contributes to the final reconstruction.

In our experiments, we captured the walls of a cluttered rectangular room with dimensions 9.84m $\times$ 7.13m with an XTion PRO LIVE sensor onto which a 3DM-GX1 MARG sensor, as illustrated by Figure 3. This room has been chosen due to the fact that it contains regions with rich geometric and color features as well as some regions that lack them, as shown in Figure 4. This has enabled us to test the robustness of the proposed approach under different conditions.

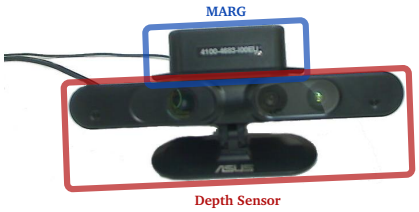


Fig. 3. Devices used during our experiments.

A. Quality of the globally optimized registration

In order to evaluate the accuracy of our methodology, we handpicked sixteen salient points from both the reconstructed map and the ground truth, and calculated the distances between all combinations of pairs in each map. The points were chosen so as to create a thorough distribution of points in the map. Figure 5 illustrates the distribution of points across the generated map.

We computed our error by subtracting the distance between two points in our map from the distance between corresponding points in the ground truth. Figure 6-a shows this error, and the normalized error (which is given by the error divided by the distance in the ground truth). Although the error behaves randomly, it tends to increase for larger distances. This insight can be useful to help us to determine the accuracy of our map as a random function of the distance between points. As depicted in Figure 6-b, which depicts the error as a percentage of the distance, we have a random variable that seems to be bound by a maximum value. This suggests that the normalized error is an independent random variable, which can be statistically shown by fitting a probability distribution function to this data with a quantile-quantile plot.

The quantile-quantile plot allows us to infer the probability distribution of a sample if the plot shows a linear data distribution. If the plot is well approximated by a linear function, then the sample can be approximated by the known distribution used in the plot. Due to the shape of the histogram given by the normalized errors, we decided to perform a quantile-quantile plot with the standard normal distribution, shown by Figure 7. As seen in the figure, a linear function can adjust well to most of he samples in our sample, with a few outliers in both

(a) Rich geometric features (b) Poor geometric features (c) Rich texture features (d) Poor texture features

Fig. 4. Different texturing and geometric conditions of the experimental setup.

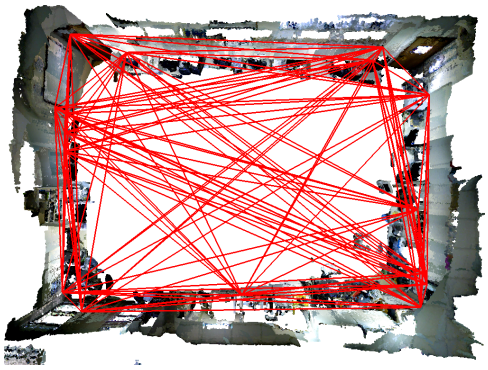


Fig. 5. Geometric features selected for comparison purposes. We calculated the distances between all pairs of geometric features, and compared the results to the distances estimated from the ground truth reconstruction.

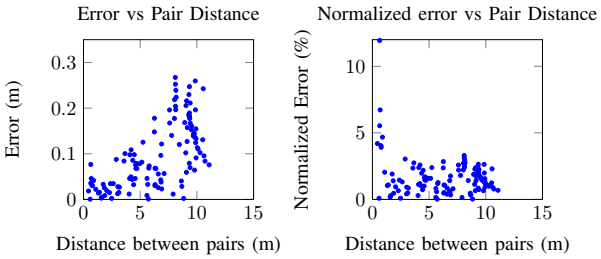


Fig. 6. Accuracy of the global reconstruction method with respect to the ground truth provided by a Zebedee sensor. The graphic on the right represents the error as a percentage of the distance between the compared points.

of its ends. Following this hint, we performed a Shapiro-Wilk test with significance level $\alpha = 0.01$, which didn't discard our approximation of the error as a normally distributed random variable. Therefore, we can say that the normalized error of the reconstruction provided by our methodology, is given by $\hat{e} = \mathcal{N}(\mu : 1.4310,\ \sigma : 1.1513)$ percent.

B. Back-end Robustness

We test the robustness of our loop closure computational block by running it on two distinct pairwise alignment methodologies. We expected that the global optimizer would receive different inputs from those alignments (a requirement that was also assessed in this phase), but would produce similar results for each of them. The pairwise alignment methodologies employed in this stage were small modifications to our Front-end block:

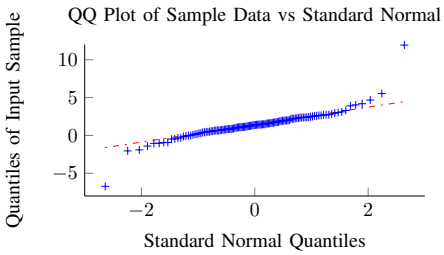


Fig. 7. Quantile-quantile plot of normalized error versus standard normal distribution.

(A) **No photo consistent alignment** The output pose from our coarse alignment is directly forwarded to the back-end block (this strategy does not perform dense registration).

(B) **No MARG-based key point matcher** The coarse alignment is not performed with input from our MARG; instead, key points are matched with a regular brute force matcher.

As it can be seen in Figure 8, pairwise alignments (A) and (B) yield different results from those obtained from the proposed methodology. We conclude that we can expect to have different alignments between adjacent keyframes even after global optimization, since our back-end performs a regular pairwise feature matcher between close point clouds, and the error between those can be too large on the results provided by strategies (A) and (B).

After performing global optimization, strategies (A) and (B) yielded maps with loop closure errors that could have been spotted with a visual inspection, as shows Figures 10. In terms of numbers, the normalized error from strategy (A) has a mean 2.62% and standard deviation 1.66 – both values are larger than the corresponding values obtained from our methodology. As for strategy (B), the data dispersion is significantly larger, with a mean of 0.14% and standard deviation 27.32. These values are shown by Figure 9.

Since the match between loop closing frames doesn't depend on the pre-alignment stage, we would expect this region of the global map to be consistent with the result obtained from our complete method, presented in the previous subsection, unless the data provided to the global alignment stage was different for both strategies (A) and (B). There is one way this could have happened: by providing the global alignment stage with

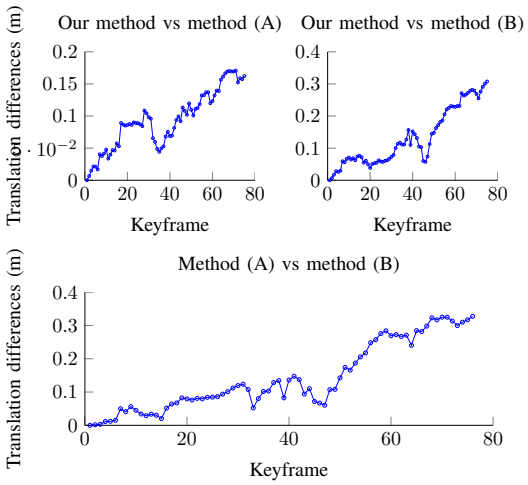


Fig. 8. Differences between key frame translations estimated by pairwise trajectories.

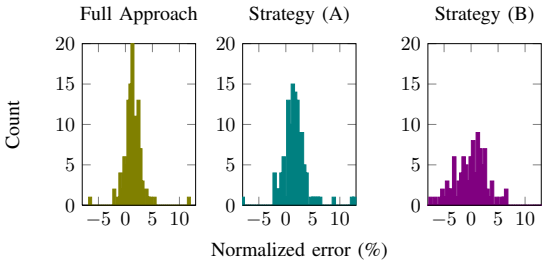
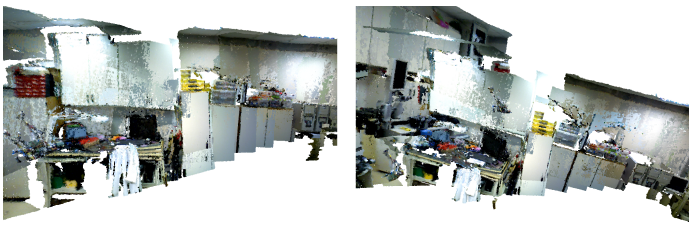


Fig. 9. Dispersion of normalized error according to the several tested methodologies.

a different set of key frames.

As can be recalled from our methodology, the key frame detection is a byproduct of the pairwise alignment stage. Therefore, different pairwise alignment strategies might yield different sets of keyframes. This was the case for both strategies (A) and (B). Wether or not this was the cause of the misalignment issues still remains to be proven.

To verify the hypothesis that the poorly closed loops by strategies (A) and (B) is a result of a different set of keyframes given to the global optimization stage, we manually changed the keyframes set of the results computed by these strategies to



(a) (b)

Fig. 10. Comparison of loop closing region as obtained from all strategies. A discontinuity can be spotted in the cabinet on the right, in (a); the same occurs in (b), with several other discontinuities that makes it the worst map estimated.

(a) (b)

Fig. 11. Map errors from pairwise alignment by strategies (A) and (B), on subfigures (a) and (b), respectively. In this experiment, the key frame ids were provided by our methodology.

be the same as that generated by our full methodology. Should our original hypothesis be correct, the expected result would be a map whose loop is consistent with the one generated by our methodology, with only differences between adjacent keyframes aligned during pairwise registration.

Strategy (A) yielded a global map visually consistent with our methodology after we used the set of keyframes detected by our approach, despite local errors introduced by its pairwise registration (also shown in the same figure). After this change, the normalized error was changed to a mean of 1.38% and standard deviation 1.64. However, it still contained misalignments introduced by its pairwise registration stage, which can be seen on 11-a.

In spite of displaying a good loop closure, strategy (B) presented several discontinuities as illustrated by Figure 11-b. These discontinuities appear between keyframes that were registered by a pairwise method, which suggests that their bonds were too weak and, therefore, were disregarded by the global optimization process. A close analysis revealed that no reliable matches were found for the 8th key frame . That is, adjacent keyframes were matched by a very small number of key points – all of them fell below our threshold of 20 key points for an acceptable match. In fact, these matches had visually protruding discontinuities. This happened because the global optimization takes into account the pairwise registration transform in order to align adjacent keyframes. Since the alignment provided by strategy (B) had a significant accumulated error at this point, the key point matching algorithm had to deal with a larger uncertainty than it was supposed to, resulting in spurious alignments with its adjacent frames.

Although this explains how different results could be obtained after global optimization, it still leaves questions as to why such results emerge. Since such differences are being observed in a region of the map where loop closure is expected to take place, our search for an explanation led us to the relaxed coarse transform algorithm, responsible for detecting loop closures and finding their respective transformations.

In a frame-by-frame analysis, we found that in several instances the ambiguity between candidates of loop closing keyframes was the cause of their incorrect alignment. As Figure 12 shows, the edge of the leftmost cabinet was vertically constant, allowing for key point matches that had a significant

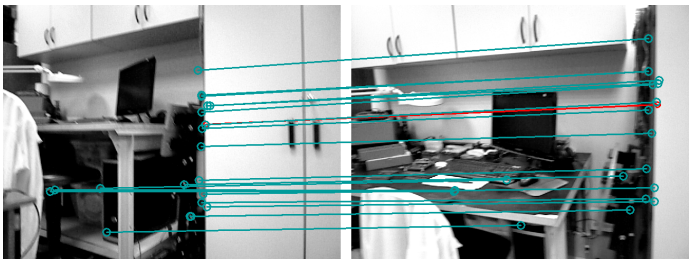


Fig. 12. Spurious key point match computed by strategy (B)

vertical error. In some other cases, similarities between the two adjacent cabinets led to a condition where several local minima could be found in which key points on the top cabinet were matched to the one in the bottom, but in these circumstances, the intersection between the frames after alignment fell below our acceptable threshold, which eliminated these bogus transforms.

C. Conditions not supported by our methodology

We know that regions with a significantly varying magnetic field would render our methodology useless, since the keypoints matching stage would be likely to find bogus correspondences, which would ultimately lead to either misalignments or even to complete divergence in pairwise registration.

Another problematic condition we have observed is that points that lie outside the depth sensor operational range (typically up to 3.5m in current commercially available RGB-D sensors) are subject to a great deal of noise and uncertainty. Such points not only introduce a large uncertainty in the coarse alignment stage, but also render this stage unfeasible if we employ a keypoint descriptor that uses normal information at each point. This happens because by introducing depth error the normals estimated at each point may vary largely.

Our methodology has been developed to reconstruct static environments or objects. This means that the subject being reconstructed must remain still when the point clouds are being acquired.

IV. Conclusions

We have presented a methodology to create globally consistent maps of static indoor environments using depth, color and inertial information. Although several works have been published aiming at the problem of environment mapping, few have tried to fuse inertial information with RGB-D data for registration purposes. With the release of many devices that incorporate inertial sensors, we expect their prices to drop quickly, making it feasible for a large amount of applications to benefit from them.

Our key point matching algorithm seeks the best correspondences at the same time it preserves Euclidean constraints. We have also used inertial data in order to discard false positives in the loop closure detection procedure.

We conducted experiments that have both validated the proposed method experimentally, and assessed the robustness of the global optimization module. We specifically studied its response to poor pairwise alignment and to ambiguities in the environment. We have also seen that, when disregarding the input from the MARG sensors, the pairwise alignment ultimately led to a map with several inconsistencies after global optimization, especially in regions with few color features.

Acknowledgments

The authors would like to thank CAPES, FAPEMIG and CNPq for the financial support provided.

References

[1] Frédéric and Bosché, "Automated recognition of 3d cad model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction," *Advanced Engineering Informatics*, vol. 24, no. 1, pp. 107–118, 2010, informatics for cognitive robots.

[2] E. Einhorn, C. Schrter, H.-J. Bhme, and H.-M. Gross, "A hybrid kalman filter based algorithm for real-time visual obstacle detection." in *EMCR*, 2007.

[3] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *PAMI*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[4] E. Einhorn, C. Schröter andter, and H. Gross, "Can't take my eye off you: Attention-driven monocular obstacle detection and 3d mapping," in *IROS*, oct. 2010, pp. 816–821.

[5] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *ISMAR*, 2007, pp. 1–10.

[6] R. A. Newcombe and J. Andrew, "Live Dense Reconstruction with a Single Moving Camera Davison, CVPR 2010." *CVPR*, 2010.

[7] D. Hyun, I. D. Yun, and S. U. Lee, "Registration of multiple range views using the reverse calibration technique," 1998.

[8] P. Biber and W. Strasser, "The normal distributions transform: a new approach to laser scan matching," in *IROS*, 2003, pp. 2743 – 2748 vol.3.

[9] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3d-ndt: Research articles," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, Oct. 2007.

[10] C. song Chen, Y. ping Hung, and J. bo Cheng, "Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images," *PAMI*, vol. 21, pp. 1229–1234, 1999.

[11] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *PAMI*, vol. 14, no. 2, pp. 239–256, 1992.

[12] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments," *IJRR*, vol. 31, no. 5, pp. 647–663, 2012.

[13] F. Steinbruecker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense rgb-d images," in *Workshop on Live Dense Reconstruction with Moving Cameras (ICCV)*, 2011.

[14] J. Stückler and S. Behnke, "Integrating depth and color cues for dense multi-resolution scene mapping using rgb-d cameras," *IEEE Int. Conf. on Multisensor Fusion and Information Integration (MFI)*, 2012.

[15] B. Huhle, M. Magnusson, W. Strasser, and A. Lilienthal, "Registration of colored 3d point clouds with a kernel-based extension to the normal distributions transform," in *ICRA*, may 2008, pp. 4025 –4030.

[16] S. Izadi, D. Kim, O. Hilliges *et al.*, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in *UIST*. New York, NY, USA: ACM, 2011, pp. 559–568.

[17] T. Whelan, J. McDonald, M. Kaess, M. Fallon, H. Johannsson, and J. Leonard, "Kintinuous: Spatially extended KinectFusion," in *RSS Workshop on RGB-D*, Sydney, Australia, Jul 2012.

[18] M. Bosse and R. Zlot, "Continuous 3d scan-matching with a spinning 2d laser," in *ICRA*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 4244–4251.

[19] M. Bosse, R. Zlot, and P. Flick, "Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping," *Robotics, IEEE Transactions on*, vol. 28, no. 5, pp. 1104–1119, oct. 2012.

[20] B. des Bouvrie, "Improving rgbd indoor mapping with imu data," Master's thesis, Delft University of Technology, 2011.

[21] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgbd mapping: Using depth cameras for dense 3d modeling of indoor environments," in *RGB-D: Advanced Reasoning with Depth Cameras Workshop*, 2010.

[22] E. R. Nascimento, W. R. Schwartz, G. L. Oliveira, A. W. Vieira, M. F. M. Campos, and D. B. Mesquita, "Appearance and geometry fusion for enhanced dense 3d alignment," *SIBGRAPI*, 2012.