Abstract—Recent increased popularity of RGB-D capable sensors in robotics has resulted in a surge of related RGB-D registration methods. This paper presents several RGB-D registration algorithms based on combinations between local visual feature and geometric registration. Fast and accurate transformation refinement is obtained by using a recently proposed geometric registration algorithm, based on the Three-Dimensional Normal Distributions Transform (3D-NDT). Results obtained on standard data sets have demonstrated mean translational errors on the order of 1 cm and rotational errors below 1 degree, at frame processing rates of about 15 Hz.

I. INTRODUCTION

In the past year the use of inexpensive sensors which provide both depth images and either RGB or intensity based images has revolutionized many aspects of mobile robotics. With the introduction of the Microsoft Kinect structured light camera, algorithms utilizing combined RGB and depth (RGB-D) information have become an increasingly important topic in mapping, navigation and perception.

Recent works present several mapping strategies, based on RGB-D sensors. Camera tracking methods, based on dense models have been shown to produce very accurate environmental models [1], based on depth information alone. Approaches to utilize both depth and intensity (or RGB) information are also being investigated, and some promising RGB-D SLAM results have recently been reported [2]. SLAM techniques today typically formulate the estimation problem into a graph structure which is then optimized with a back-end, for example, the g2o framework [3]. Essentially two types of constraints are posed into the graph: consecutive estimates (frame-by-frame) also called visual odometry and constraints imposed by loop-closure. Consecutive frames are commonly assumed to have a rather small change in pose, which is successfully utilized in [4] where the authors obtain accurate tracking of the relative pose by minimizing a reprojection error function between the consecutive frames.

For detecting loop closures, visual features, for example SURF [5], and vocabularies [6] to describe frames are commonly used.

This work intends to be suitable both for obtaining consecutive frame-to-frame registration estimates at high frame rates and for detecting loop closure events. The algorithms presented here start from a similar vantage point as the RGB-D SLAM approach presented in [2], but focus solely on the frame-to-frame pose estimation and do not perform graph relaxation. The initial visual feature based estimates are then used to seed a recently proposed fast 3D registration algorithm [7] and obtain refined and improved tracking performance. An uncertainty or covariance estimate of each estimated constraint is also directly obtainable from the registration method. Due to the high frame rate of RGB-D sensors, many methods (including the one presented here) attempt to achieve real-time performance, which for vision applications is referred to be at least 10 Hz [8].

The next section presents our proposed combined visual and geometric registration method. Section III then presents results obtained on some of the TUM RGB-D data sets from [9] and preliminary comparisons with previously reported results on the same benchmark suite [4]. Finally, we discuss some directions for future improvement of the proposed methods.

II. APPROACH

The suggested approach assumes that both an RGB (or intensity) image $I_{RGB}$ and a depth image $I_{D}$ are acquired at approximately the same time $I = \{I_{RGB}, I_{D}\}$. The main idea is to utilize standard vision techniques (local visual features) to find corresponding regions from $I_{t}$ and $I_{t+1}$ and then use RANSAC to find a consistent alignment. The geometrical information in the vicinity of the selected features can then be efficiently utilized as an input to a fast 3D registration method.

A. Visual Features Registration

For each image $I_{RGB}$ a set of visual features $F$ are extracted. To find the correspondences between features from two images $I_{t}$ and $I_{t+1}$ a brute force search is done between image feature $F_{t}$ and $F_{t+1}$, where the best associations are kept as a possible set of candidates matches $C$. Every feature point $f_i$ in $C$ is then associated to a neighbourhood of pixels in the depth image $I_{D}$. The pixels in each neighbourhood can further be represented as a set of corresponding 3d points. The average 3D point for each neighbourhood can then be viewed as the geometrical position of the feature point $f_i$. By choosing a neighbourhood size of one pixels, a direct point-to-point correspondence can be established. On the contrary, choosing larger neighbourhood size (or support size) leads to averaging of the feature position and a point-to-region correspondence. Finally, the points in each feature neighbourhood are stored and later utilized in the geometric registration step.

The set of candidate matches $C$ and the set of feature point 3D positions can be used to establish a least square error transformation. However, due to local visual imilarity, a large portion of outliers and wrong associations can be
expected. Thus, a 3 point RANSAC algorithm is deployed as a consistency filtering step. Three feature pairs from $C$ are randomly selected and used to determine a closed form solution for rotation and translation [10]. The translation and rotation estimate are then used to transform all feature points in a common coordinate system. The number of transformation inliers — associated matching features that are close to their corresponding feature points, is used as a ranking criteria, in order to select the best consistent transformation. Finally, the transformation obtained is used as an initial estimate in the geometrical registration step.

B. Geometric Registration using the 3D-NDT

Several 3D geometric registration approaches could be used to successfully perform fine registration and adjustment of the obtained poses. Local registration methods, such as the Iterative Closest Point (ICP) method [11], are often used as a post processing step for feature-based registration methods (See for example FPFH features and a non-linear Levenberg-Marquardt registration [12]). However, even state of the art algorithms like Generalized ICP [13] can be too slow to ensure real time operation, due to the large number of points that need to be considered. The 3D-NDT (Normal Distributions Transform) is a spatial representation that reduces the number of components in a point cloud by an order of magnitude, by locally representing space as Gaussian probability density functions. The 3D-NDT has previously been used for registration, using a Point-to-Distribution [14] semantic — maximizing the probability of a set of points, given the probability distributions in a 3D-NDT model of a known point set. In a recent article, Stoyanov et al. [7] propose to register directly two 3D-NDT representations (Distribution-to-Distribution or 3D-NDT-D2D), thus maximizing the overlap between two sets of Gaussian pdf.

In principle, any 3D geometric registration algorithm could be used to refine the initial alignment, obtained using visual features and RANSAC. The choice of the 3D-NDT-D2D for this task is motivated not only by the fast runtimes, but also by its suitability to a particularly relevant modification. Instead of computing the 3D-NDT representation of the full depth images, we propose to only consider the immediate local neighbourhoods of each of the detected local visual feature points. Thus, the number of Gaussian components can be decreased substantially and consequently, a full 3D point cloud corresponding to the depth image need not be computed. Several options for performing the 3D-NDT registration were explored and are further described in the following subsection.

C. Tested Algorithm Variations

Several options for performing the combined visual and geometrical registration exist and were explored. Figure 1 shows a schematic explanation of the different registration method variations tested. All algorithms start from a common input — two consecutive RGB-D images $I_1, I_2$. The first option tested is labeled RANSAC’ and consists of just two major blocks — extracting and matching SURF features and running RANSAC to find a consistent transformation between the 3D coordinates of the feature points. The second option is the main RANSAC variation considered, which performs an additional step — the 3D coordinates of the feature points are calculated as the mean position of all
points in the feature neighbourhood. The first, baseline 3D-NDT-Feature registration algorithm \((\text{NDT}^F)\) keeps track of not only the mean, but also the covariance of each feature neighbourhood. In this way each RGB-D Image \(I\) is represented as a set of Gaussian distributions around feature points. \(\text{NDT}^F\) performs a 3D-NDT-D2D registration step on these Gaussian distributions, taking into consideration the correspondences and transformation found by RANSAC. The two variations \(\text{NDT}^F_2\) and \(\text{NDT}^F_3\) test two possible noise reductions for features, during the RANSAC step — namely a statistical outlier removal at a 90\% significance and a distance-based feature rejection respectively. The \(\text{NDT}^F_4\) variation tested skips the RANSAC and correspondence selection steps altogether and directly performs a registration on the distribution sets. Finally, a standard 3D-NDT-D2D registration is performed for comparison, using a regular grid 3D-NDT model of the full point clouds, computed from the depth images. An example registration of two consecutive frames using the RANSAC and \(\text{NDT}^F\) algorithms is shown in Figure 2.

III. EVALUATION AND RESULTS

To obtain real-time performance (>10Hz), the RGB images were first downscaled with a factor of \(\frac{1}{16}\) (scale down four times both width and height). The feature detector used was the standard SURF implementation in OpenCV [15]. All evaluations were performed using a single core with 4800 bogomips (Intel Q6600). The evaluation utilizes the RGB-D data sets collected by Strum et al. [9]. Results from the method proposed by Steinbrücker et al. and Generalized ICP (GICP) are taken from the paper [4].

Table I summarizes the accuracy performance of the different evaluated algorithms on several of the standard RGB-D benchmark sets. The reported results show the mean translational errors over all frame pairs, as calculated using the ground truth path and evaluation tool, provided in the benchmark. We note that the performance of the visual features / RANSAC combination and the 3D-NDT-D2D depth-based registration is comparable over all the data sets. Both algorithms have average translation errors on the order of 2-3 centimeters. Next, we note that three of the combined algorithms - \(\text{NDT}^F_2\), \(\text{NDT}^F_3\), and \(\text{NDT}^F_4\) produce comparable results, with an error of about 1.5 centimeters, or half of the error of the previous two algorithms. The difference between the performance of these three variations will be discussed in subsequent paragraphs. Finally, the variation \(\text{NDT}^F_4\) which relies on the registration of only the geometrical information around each visual feature, performs substantially worse, with mean errors of several centimeters. The lack of correspondences and the very sparse models used result in local minima problems for the registration method in this case and need further investigation.

Table II summarizes the runtime performance (in seconds) of each of the proposed algorithms over all the data sets. Again, the runtime performance of the first three \(\text{NDT}^F\) variations is very similar, with a mean of about 15 Hz. \(\text{NDT}^F_4\) performs much faster, due to the sparsity and lack of correspondence searches, but the obtained accuracy is suboptimal, as already discussed. The RANSAC visual algorithm is marginally faster then the \(\text{NDT}^F\) approaches, totaling around 0.05 seconds per frame. The full 3D-NDT-D2D registration algorithm runs a couple of times slower, with around 0.15 seconds per frame, which is still two orders of magnitude faster then the geometrical registration algorithm (GICP), used as a baseline by Steinbrücker et al.

Looking at the results in Tables I and II, the performance of the two modified versions \(\text{NDT}^F_2\) and \(\text{NDT}^F_3\) does not appear to differ much from that of the standard \(\text{NDT}^F\) version. Some additional tests were performed to investigate this similarity in performance. Figure 3(b) shows how the mean translation error of \(\text{NDT}^F_2\) varies for all data sets when the feature cutoff threshold is changed. The idea behind

Fig. 2. An example registration of two consecutive frames, using the RANSAC (in (a)) and \(\text{NDT}^F\) (in (b)) algorithms. Problematic areas are highlighted.
cutting off far-away features is motivated by the severe degeneration in accuracy of the Kinect sensor at distances above 3.5 meters. The results however seem surprising at first — using a cutoff threshold does not offer an improvement in any of the cases, and in fact can only make results worse. The error naturally increases if the cutoff threshold used is too low — lower or equal then 3 meters for most data sets, due to a depletion of feature points. Using a higher cutoff threshold however has no overall effect. Two possible explanations for this effect are the low percentage of feature points at long distances (in the considered data sets) and the general robustness of RANSAC in selecting a well fitting transformation in the presence of a high number of outliers. Similarly, the possible improvements due to trimming the RANSAC matches to only use 90% of inliers in the transformation computation have little effect on the subsequent 3D-NDT-D2D registration performance.

The impact on the support size - the surrounding region around each keypoint used to calculate the 3D-NDT distribution are depicted in Fig. 3(a). To obtain a resonable distribution it is important that the the size is not to small. All other results presented utilize a support size of $15 \times 15$ pixels.

Table III also demonstrates the performance of two of the key algorithms — RANSAC and NDT$F$ without a mean neighbourhood filter. The approaches, labeled RANSAC" and NDT$F_2$ set the 3D position of each keypoint directly to the value calculated from the depth image. Thus, when keypoints are detected in noisy regions or close to edges in the depth image, the associated 3D positions can vary substantially in between different frames. Although the results obtained show a worse performance for the unfiltered keypoint versions of both algorithms, the performance drop is not very significant, testifying yet again to the robustness of the 3-point RANSAC method used.

Some measure of a baseline comparison with state of the art approaches can be obtained through studying the results of Steinbrucker et al. [4]. Another interesting point of comparison would be with the open loop registration part of the RGB-D SLAM algorithm [2], but unfortunately only values
This paper presented several RGB-D registration methods based on a combination between local visual features and 3D-NDT registration. The implementation of all variants presented here is available as a ROS package (ndt_feature_reg) at http://code.google.com/p/oru-ros-pkg/. The results obtained on standard data sets have demonstrated a promising performance, with mean translational errors on the order of 1 cm and rotational errors usually below 1 degree. This performance is attained in near real-time, with a frame processing rate of about 15 Hz. Future work will concentrate on lowering the rotational errors obtained and identifying local minima in the registration, as well as applications as a registration front-end in a SLAM system.

ACKNOWLEDGMENTS

This work has been partially supported by the European Union FP7 Project “RobLog — Cognitive Robot for Automation of Logistic Processes”.

REFERENCES


for the absolute error after loop closure have been reported. Table IV shows a comparison between the NDTF algorithm and the Generalized ICP and visual odometry approaches. The mean translation errors obtained by the proposed approach are clearly less accurate than the visual odometry, but similar to the ones obtained by Generalized ICP. We note however the relatively large differences between the mean and median values of our approach, which testifies to the presence of some large outliers. Indeed, examining the errors obtained for a single dataset (freiburg2desk) shown in Figure 4, we can detect several clear outliers in both translational and rotational errors. Identifying such outliers online should be possible by examining the shape of the 3D-NDT-D2D objective at the (local) minimum and will be one important future direction. Another important future direction for improvement will be the investigation and reduction of the (relatively) large rotational errors obtained by our approach.

Finally, we demonstrate the stability of the proposed approach (NDTF is used here) to skipping of frames. Figure 5 shows a plot of the mean translation error obtained, depending on the number of frames skipped — i.e., the result at n on the x axis of the plot is obtained by registering every n-th frame. Naturally, the accuracy of the algorithm degenerates with increasing the number of skipped frames, but the rate of drop in performance is relatively low. The performance is naturally influenced by the overlap between frames and the speed of movement of the sensor, but the preliminary results promise that in the future applications featuring faster camera motions or using less processing power will be possible.

### TABLE III

<table>
<thead>
<tr>
<th>Dataset</th>
<th>x (m)</th>
<th>y (m)</th>
<th>θ (deg)</th>
<th>fps (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-360</td>
<td>0.014</td>
<td>0.010</td>
<td>0.592</td>
<td>0.483</td>
</tr>
<tr>
<td>1-desk</td>
<td>0.016</td>
<td>0.012</td>
<td>1.009</td>
<td>0.820</td>
</tr>
<tr>
<td>1-floor</td>
<td>0.015</td>
<td>0.007</td>
<td>1.027</td>
<td>0.402</td>
</tr>
<tr>
<td>1-room</td>
<td>0.011</td>
<td>0.008</td>
<td>0.635</td>
<td>0.502</td>
</tr>
</tbody>
</table>

### TABLE IV

<table>
<thead>
<tr>
<th>Dataset</th>
<th>RANSAC</th>
<th>RANSAC</th>
<th>ORD</th>
<th>ORD'</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-360</td>
<td>0.035</td>
<td>0.035</td>
<td>0.014</td>
<td>0.015</td>
</tr>
<tr>
<td>1-desk</td>
<td>0.021</td>
<td>0.022</td>
<td>0.016</td>
<td>0.019</td>
</tr>
<tr>
<td>1-floor</td>
<td>0.024</td>
<td>0.025</td>
<td>0.016</td>
<td>0.017</td>
</tr>
<tr>
<td>1-room</td>
<td>0.020</td>
<td>0.023</td>
<td>0.015</td>
<td>0.018</td>
</tr>
<tr>
<td>2-desk</td>
<td>0.025</td>
<td>0.025</td>
<td>0.011</td>
<td>0.010</td>
</tr>
</tbody>
</table>

IV. DISCUSSION

This paper presented several RGB-D registration methods based on a combination between local visual features and 3D-NDT registration. The implementation of all variants presented here is available as a ROS package (ndt_feature_reg) at http://code.google.com/p/oru-ros-pkg/. The results obtained on standard data sets have demonstrated a promising performance, with mean translational errors on the order of 1 cm and rotational errors usually below 1 degree. This performance is attained in near real-time, with a frame processing rate of about 15 Hz. Future work will concentrate on lowering the rotational errors obtained and identifying local minima in the registration, as well as applications as a registration front-end in a SLAM system.
Fig. 4. Figure 4(a): translation error (left) and Figure 4(b): orientational error using NDTF with \textit{freiburg2desk} data set.

Fig. 5. Translation error while skipping intermediate frames. Figure 5(a): \textit{freiburg1desk} data set. Figure 5(b): \textit{freiburg2desk} data set.
