2013 European Conference on Mobile Robots (ECMR)
Barcelona, Spain, September 25-27, 2013

Adaptive Iterative Closest Keypoint

Johan Ekekrantz[1], Andrzej Pronobis[1], John Folkesson[1] and Patric Jensfelt[1]

Abstract— **Finding accurate correspondences between overlapping 3D views is crucial for many robotic applications, from multi-view 3D object recognition to SLAM. This step, often referred to as view registration, plays a key role in determining the overall system performance. In this paper, we propose a fast and simple method for registering RGB-D data, building on the principle of the Iterative Closest Point (ICP) algorithm. In contrast to ICP, our method exploits both point position and visual appearance and is able to smoothly transition the weighting between them with an adaptive metric. This results in robust initial registration based on appearance and accurate final registration using 3D points. Using keypoint clustering we are able to utilize a non exhaustive search strategy, reducing runtime of the algorithm significantly. We show through an evaluation on an established benchmark that the method significantly outperforms current methods in both robustness and precision.**

I. INTRODUCTION

Sensor data is essential to robotics. More sensors are producing better data at faster rates. At the same time the computational power of a robot is limited, rendering the robot unable to utilize all the data it has collected when making control decisions. That data is therefore put into a representation that allows it to be more easily used. This invariably require so called data registration or placing the data taken at different locations into the same frame of reference. The focus of this paper is data registration between RGB-D frames.

A lot of research has been done on data registration of 2D and 3D range data captured by laser scanners and sonars. The same is true for RGB camera image registration. Since the introduction of cheap RGB-D cameras in 2010 these sensors have become very popular in the field of robotics. The combining of RGB data with the depth, 'D', range data in one sensor introduces new challenges and new opportunities. The data registration problem is equivalent to finding the transformation between the sensor poses for each frame of sensor data.

For range data, either 2D slices or 3D point clouds, the most widely used method is the iterative closest point, *ICP*, algorithm [1]. This method requires an initial guess for the transformation and will not converge correctly if started too far away. ICP works on sets of geometric points. It requires only the locations of the points relative to the sensor in each frame. From this it returns the transformation between the two sensor frames.

For RGB image data the registration can be done using image features or keypoints. The process is to first detect points of interest in the image. Then (optionally) compute a descriptor from the context (i.e. neighboring pixels) of the point. Registration is done by matching these 'keypoints' either by using the descriptors or the geometric constraints or both. As the camera only gives a bearing to the points the geometric constraints are more complicated than for range sensors. Having RGB-D sensors allows us to both use the simpler geometric registration and the selective power of the RGB descriptor.

In this paper, we present Adaptive Iterative Closest Keypoint (AICK), a registration algorithm for RGB-D views which builds on the idea of ICP. The algorithm preserves the accuracy of ICP for small transformations, while providing a drastic improvement of robustness to larger view rotations and translations without the need for an initial guess given sufficient overlap between the frames. Our algorithm exploits both depth and visual information and relies on keypoints detected in images associated with 3D positions in the local reference frame and a visual descriptor. The key property of the algorithm is the ability to weigh the importance of the visual descriptor and the 3D position while iteratively optimizing the transformation. This allows us to exploit the distinctiveness of appearance features for improved initial robustness and accuracy of point locations for the final precision. In addition to this we investigate how one might relax some of the more computationally expensive parts of the algorithm without unduly sacrificing the quality of the registration. Our aim is an algorithm for real-time systems where both low computational load and high performance is a requirement.

We compare the proposed method to generalized ICP (GICP) [2], the 3D normal distribution transform (3D-NDT) [3] and a method based on RANSAC [4] using keypoints which we will call 3-point RANSAC. We perform an extensive evaluation of the four algorithms on a publicly available dataset[1] [5] and employ an established benchmarking procedure and performance measure [5]. The results show improvements in both robustness to large initial transformations and accuracy of the final result. Furthermore, our method converges fast and requires significantly less computational power making it a great choice for real-time applications.

In the remaining parts of the paper, we first provide an overview of registration methods. Section III provides details of the proposed algorithm. Section IV covers the setup for the experimental evaluation. Finally, we present the results of the experimental evaluation in Section V.

[1]The authors are with the Centre for Autonomous System at KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden {ekz,pronobis,johnf,patric}@csc.kth.se

[1]http://vision.in.tum.de/data/datasets/rgbd-dataset

978-1-4799-0263-7/13/$31.00 ©2013 IEEE

II. RELATED WORK

Methods of data registration can be subdivided by the sensor suite that is being used. A large body of work covers image registration for cameras [6] where the data points represent bearing only information. When motion sensors are available and long sequences of frames are being handled together, data registration is often called simultaneous localization and mapping (SLAM), structure from motion or bundle adjustment. We restrict ourselves here to range and bearing data registrations between two frames of data.

The methods that are most related to our work are based on the ICP algorithm. ICP was introduced to the registration context in [1], [7]. In these, early but still very relevant, works the closest point in the Euclidean sense forms the data association pairs and the sum of the squared distances between the matched points is the goodness of fit measure. The ICP algorithm is bootstrapped using some estimated initial transformation. This is then followed by alternating between finding the data association, i.e., closest point pairs and finding the transformation that minimizes the goodness of fit measure. This is repeated iteratively until convergence. The primary advantage of ICP is its simplicity. Notice that in this original version no consideration was made for noise in the sensors. Neither the fact that the actual scan points are sampled from continuous surfaces and thus the two data sets will not include the exact same points but rather nearby points. This can be thought of as a type of discretization noise.

There has been a number of refinements to the point to point ICP approach that address the issue of discretization noise. In [8] points are compared to planes so that errors are projected on the surface normals. In this way it is theoretically possible, in the absence of sensor noise and surface curvature, to form a perfect fit. In [2] the GICP algorithm is introduced. Here both data frames are modeled as being sampled points from planar surfaces. This method is currently popular and considered to give overall good performance.

There has also been work on variations of ICP that take account of the stochastic processes in the data acquisition. In [9] and later extended to 3D in [3], [10] the normal distribution transform (NDT) is developed. These explicitly models the probability of sampling a point near the surfaces. A similar probabilistic approach is taken in [11]. These methods make no explicit data association as all pairs contribute an amount weighted by their probability.

Metric-Based ICP[12] is a version of the ICP algorithm which is designed to be able to deal with large angular differences. This algorithm makes a change to the distance metric (i.e. goodness of fit measure) to account for the effect that points far away get large displacements due to rotations.

Given recent increased availability of RGB-D sensors new variations of the ICP algorithm has been designed to perform well on these sensors. [13] proposed a version in which the distance metric was changed to not only measure Euclidean distance between points but also take into considerations a color value (hue) to obtain better matching. In [14] omnidirectional stereo cameras were used to form 3D point clouds with color. The ICP algorithm was then modified to use the color information in the metric in what they call color ICP.

A variation on ICP called Multi-scaled EM-ICP is presented [15]. They do not assume one data association but rather consider a weighted combination of matches with the scale setting the weight. This shares some properties with AICK in that they suggest that the scale can be changed adaptively as one iterates the method thus avoiding local minima.

In [16] the ICP distance function is modified to include a weighted sum of the Euclidean distance and the feature distance. The method employed features based on moments, spherical harmonics and curvature of a global density function that approximated the surface geometry. The work is similar to our approach in the fact that visual descriptors are employed to enhance matching. However, we differ from this method in the choice of features and most importantly in the way we automatically vary the influence of visual information during convergence providing both robustness and precision. A review of the many variations of range image registration methods is given in [17]. In view of that study, our method can be seen as a combination of point signature coarse registration and ICP fine registration where the same algorithm handles both aspects.

The most computationally expensive part of ICP is typically finding the closest points. This has a complexity of $\mathcal{O}(N^2)$ in a naive implementation. A common way to speed this up is to use a kd-tree (or a set of trees) which reduces the complexity to $\mathcal{O}(Nlog(N))$.

In [18] the Kinect Fusion algorithm is presented. It uses a dense, non-parametric, representation for the reference frame from which an artificial point cloud is sampled and registered against. With streams of consecutive frames Kinect Fusion integrates every new frame into a larger reference frame to which new frames are registered.

In a track parallel to the work with lasers, vision based systems have achieved impressive results. Also here the registration problem is a key issue. Using key points reduces the need to treat all pixels and using feature descriptors allows for reliable associations. ICP can be used to register RGB images by using keypoints as shown in [19].

SIFT [20] and later SURF[21] are still widely used and give very good results. While being discriminative and somewhat invariant to scale and rotation SURF and especially SIFT are relatively expensive to compute. Several simpler but faster to compute features have been suggested such as BRIEF [22] and later ORB [23] which extends BRIEF with invariance to rotation. The detection of keypoints is often done by FAST [24] or Harris corners [25]. Additional recent descriptors include BRISK[26] and FREAK [27]. In [28] a system for adaptively extracting key points from a RGB-D video stream and matching them to do motion estimation of the camera, so called visual odometry, is presented.

A common data association problem is that of looking for a match between one frame and all frames previously seen.

Finding these, so called, loop closures are key to a successful implementation of SLAM.

Here the question is first if the two frames match at all and if so what the transformation is. While it is often technically possible to limit the search space for the possible matching frames, one reduces robustness by relying on the position when looking for matches. Matching feature by feature in each frame is prohibitively slow. A common approach taken is to make use of visual vocabularies [29]. The basic idea is to form clusters in descriptor space and assign a label to each cluster or word. The discretisation of descriptors into words means that feature matching can be done by comparison two integer indices (the label of the word). An image can be described by a set (bag) of words and represented by a histogram counting the number of times a certain word occurs in the image. Matching two images has then been reduced from a $\mathcal{O}(N^2)$ matching operation where high dimensional descriptors are compared to a constant time operation of comparing two histograms. This has laid the foundation for FAB-MAP [30] and its follow-ups.

A major part of registration is the problem of outlier rejection i.e. the fact that there may be regions with no overlap. Using a suitable model, RANSAC [4] can be used to separate inliers from outliers and calculate model parameters.

III. THE AICK ALGORITHM

The input for the AICK algorithm is a set of invariant keypoints detected in each RGB-D frame. Each keypoint is associated with a 3D position in a local reference frame and an appearance descriptor. The keypoints are generally selected from a large set of possible points. AICK require these keypoints to be stable, meaning that the same point should be detected as a keypoint in several consecutive frames. The AICK algorithm does not make any assumptions about the keypoint detector/descriptor algorithm employed however it is desirable for the keypoint descriptors to be both rotation and scale invariant and preferably be unaffected by illumination. In this work we have evaluated both SURF [31] and ORB [23] keypoints.

The original ICP algorithm finds data associations purely based on the position of points. First, it computes the Euclidean distance, d_e, between all pairs of points, where one point comes from the point cloud A and one from the point cloud B. Then, for each point in point cloud A, it selects the point from point cloud B with the smallest distance.

A. Adaptive distance metric

In the presented algorithm, the data association is formed by taking into account both location and visual similarity. In AICK only the keypoints, and not all points in each 3D data frame, are used. We replace the Euclidean distance d_e with a weighted sum of d_e and the distance between the two keypoint descriptors d_d:

$$d(a,b) = (1 - w(i))d_e + w(i)d_d, \tag{1}$$

where the d_d equals the squared L2 norm of the difference in descriptor vectors scaled by a constant depending on the type of feature descriptor used, i being the iteration numbers $i \in \{0, 1, 2, ...\}$ and $w(i) \in (0, 1)$ being the weight given to the feature distance. We define $w(i) = \alpha^i$ where α acts as shrinking factor, making $w(i)$ decrease exponentially with respect to i.

Introducing a weighted measure is of a key importance for the adaptiveness of our algorithm. By our choice of w, the Euclidean distance factor is completely neglected for the initial match ($w(i = 0) = 1$). As a result, the algorithm does not require an initial guess for the transformation between the point clouds. Starting from the second iteration, we allow matches which are geometrically close ($w(i) < 1$), but perhaps not the closest in appearance. This continues until the descriptor distance is given very little weight as i becomes large, i.e. we essentially no longer consider appearance. At this point, the original ICP is performed for the fine registration.

One of the challenges that arise with the new definition of $d(a, b)$ is setting the threshold on the distance to identify points that are present in A, but not in B. In practice, the numerical values of d_e and d_d can differ by an order of magnitude. Thus, the scale of the distance varies with i. For that reason we use the following criterion

$$d(a,b) \geq (1 - w(i))\lambda_e + w(i)\lambda_f, \tag{2}$$

where λ_e and λ_f are two separate threshold parameters, one that matches the spatial scale and one that matches the descriptor distance scale. This is completely equivalent to scaling the descriptor distance by λ_e/λ_f and using λ_e as a constant threshold for all iterations.

B. Non exhaustive search strategy

Experiments with the AICK algorithm show (see Section V) that the performance is high even when not all keypoints are used. If we limit the search for the keypoints in one frame to only a small subset we might miss some matches. However, as the experiments will show this does not matter given that we start with enough keypoints. This opens up ways to make the algorithm more efficient by trading off the expensive step of finding all the matches that fall below our threshold.

To do this we use the method of learning a 'vocabulary' of words as in the bag of words method.[2] The learning step is based on different 'training' data than the registration is done on. Learning is essentially clustering the descriptors from all the training images into a predetermined number of clusters. Then the words of the vocabulary consist of the mean descriptors for each cluster.

We associate every keypoint, p_k, to a list of its closest words, which we denote as $\Psi(p_k)$. $\Psi(p_k)$ contains the words to which the descriptor distance of the keypoint is less than a threshold, R_w. This can be done swiftly if the vocabulary contains few clusters or if the words are arranged in a tree structure that speeds up this search. It is important to note

[2]We do not use the 'bags' in this work only the words. The bags might be useful to chose which two frames to try to register to one another which is a question not addressed here.

that this is only done once per frame, i.e., if we match the frames to many other frames we need not recompute Ψ.

When searching for the closest match to a particular keypoint in one frame we use only its closest word. We then search the other frame's $\Psi(p_k)$[3] for all occurrences of that word and compare to the corresponding keypoints. The result of this is that instead of having to match all points to all points we only match each point to a (small) subset of the points in the other frame.

As we will show this can speed up the expensive association step by an order of magnitude in most cases. We consider this a generalization of the original algorithm as using $R_w = \infty$ is equivalent to the original algorithm.

As the descriptor space is vast, most of the space will not be near any word. We are therefore relying on the words being a good representation for most of the common keypoints that are found in images. We expect that the keypoint clusters will be mostly tight and few points will be far from any word.

IV. EXPERIMENTAL SETUP

In order to evaluate our method we employed a publicly available dataset designed for the purpose of benchmarking RGB-D SLAM algorithms in realistic indoor environments [5]. The dataset contains sequences of RGB-D data captured using the Kinect sensor along with ground truth acquired with a motion capture system. We use a part of the dataset captured in an office space with the RGB-D sensor handheld. In particular, we use the sequence *fr1/room* which at the time of writing this paper was the longest of all the sequences in the natural office environment subset (see fig. (1) for sampled images from the dataset). This data set is well suited to its designed purpose of testing state of the art registration algorithms in that the motion has all 6 degrees of freedom and the movement is both rapid and uneven. Fig. (2) shows an example sequence undergoing rapid motion, which is indicated by the presence motion blur.

A. Performance Measure

In order to enhance comparability of the results, we employed a performance measure provided together with the dataset [5]. The measure is based on the relative pose error, which is found by first transforming the origin pose using the estimated transformation and then transforming it back using the inverse of the ground truth transformation. In a perfect case without error, this results in a pose matching the origin pose.

$$E_i = G_i^{-1} Q_i - I, \tag{3}$$

Where G_i is the ground truth transformation for transformation i, Q_i is the estimated transformation and I is the identity matrix. As suggested in [5] we analyze the translation component of E_i by measuring the relative distance between the pose obtained after the two transformations described above and the origin pose. This error will be given in meters, see eq. (4) for mathematical formulation. As a means of summarizing the results for a set of translation errors we define *success ratio* as the ratio of translation errors smaller than some threshold λ_t in the set. That is the registration is considered a 'success' if it satisfies eq. (4).

$$E_i^{Translation} = (\sum_{j=0}^{2} ||E_{i,j,3}||^2)^{1/2} < \lambda_t. \tag{4}$$

It is worth noting that when *success ratio* $= 0.5$, λ_t is the median error. Similarly to using the median error the *success ratio* considers all outliers as equal, meaning that

[3]By creating an index per frame from words to keypoints based on the $\Psi(p_k)$ we can find the points to compare quickly.

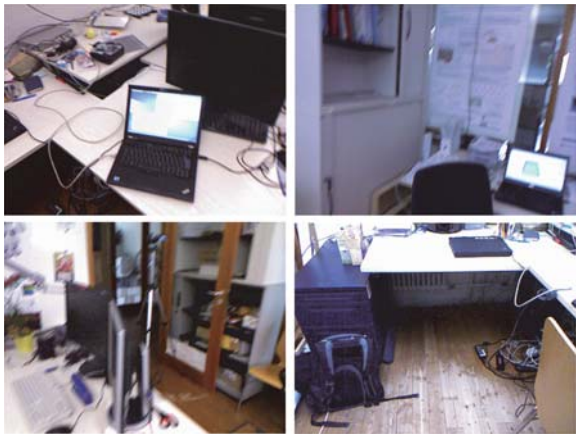

Fig. 1. Sample views from the test set.

Fig. 2. Consecutive frames captured at 30fps from the test sequence exemplifying the rapid movement of the handheld camera.

gross outliers does not bias the analysis. This formulation allows us to analyze the distribution of errors by varying the threshold λ_t.

B. Algorithms tested

We ran and compared three different registration algorithms in addition to AICK[4] on the test set. The parameters for the algorithms were optimized by hand by testing a large set of values to yield good performance within a maximum of roughly five minutes of execution time per pairwise registration.

1) GICP: We use the GICP implementation provided by the Point Cloud Library (PCL [32])[5].

2) 3D-NDT: We use the 3D-NDT implementation provided by the Point Cloud Library (PCL [32])[6].

3) 3-POINT RANSAC: We used the RANSAC algorithm on this problem by first forming a list of potential matching keypoint pairs based on the similarity of the descriptors only. We then randomly select three of these pairs to define a transformation between the frames, which we will call the 'model'. We then count the number of 'inliers' according to the model. The model with the most inliers is chosen and updated by using all of the found inliers. In forming the list of potential matched pairs only associations between keypoints with descriptor distance $d_d \leq \lambda_f$ are used. Inliers are calculated by transforming the keypoints in one frame by the model and associating the transformed keypoints to the closest keypoint in the other frame. If the euclidean distance $d_e \leq \lambda_e$ between these keypoints the association is counted as an inlier [7]. For the 3-point RANSAC algorithm we use SURF keypoints.

We will use two different types of keypoints, SURF [21] and ORB [23]. The Surf keypoints will be extracted using OpenSURF Library[33][8]. Using our test set we found an average of 906 surf keypoints with valid depthdata in an average of 0.12 seconds. To extract the ORB keypoints we use OpenCV [34][9]. Using our test set we found an average of 857 ORB keypoints with valid depthdata in an average of 0.011 seconds.

C. Experimental Procedure

We performed registration experiments by estimating transformations between consecutive frames of the data sequence. In order to test robustness to larger transformations, we performed the experiments for pairs of frames separated by different lengths of time. Performance is measured

[4]AICK using $\lambda_e = 0.01m$ and $\lambda_f = 0.2$.

[5]GICP was allowed to run for 25 iterations. Rejection threshold $= 0.004m$.

[6]To keep the runtime reasonably low the pointclouds were subsampled through the use of a voxelgrid with a voxel size of $0.02m$. 3D-NDT was allowed to run for 25 iterations, with $resolution = 0.1$ and $stepsize = 0.09$.

[7]We iterated the RANSAC over 400 random models in searching for the best model using $\lambda_e = 0.02m$ and $\lambda_f = 0.2$.

[8]With $upright = true$, $octaves = 5$, $intervals = 5$, $init\_sample = 2$ and $threshold = 0.00001$

[9]With $nfeatures = 1100$,$scaleFactor = 1.2$, $nlevels = 8$, $edgeThreshold = 2$, $firstLevel = 0$, $WTA\_K = 2$, $scoreType = ORB :: HARRIS\_SCORE$, $patchSize = 31$

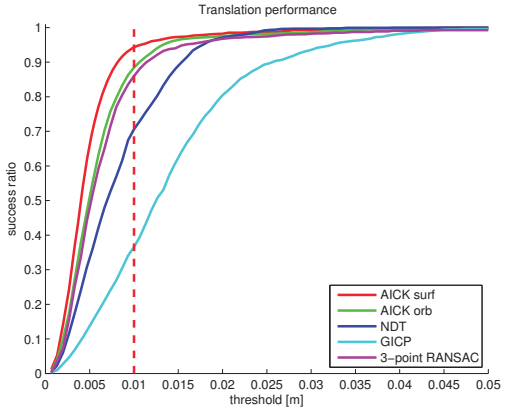

Fig. 3. The *success ratio* as a function of the threshold on the translation error in m. Here we use all the found keypoints. The red dashed line shows the threshold used in fig. (4). Meaning that the intersections with the red dashed line are equivalent to values for the *success ratio* in fig. (4) when the time difference between frames equals 30 ms.

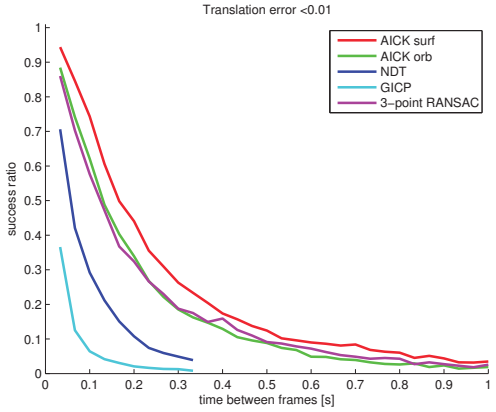

Fig. 4. *success ratio* as a function of the time difference between frames with a fixed threshold on the translation error of 0.01 m. Here we use all the found keypoints.

quantitatively using the measure described in Eq. 4. The point clouds were created with calibrated camera parameters. In section V-B we visualize the effects of accumulating a sequence of consecutive frame transformations and transforming the appropriate pointclouds into a common coordinate frame.

V. EXPERIMENTAL RESULTS

In fig. (3)[10] we plot the *success ratio* versus a varying λ_t (of eq. (4)) up to 0.05 meters using consecutive frames (around 30ms apart) for the different algorithms. This allows us to see both the size and variation of the translation error of the different methods when the transformation between frames is relatively small. A steep curve can be interpreted as good performance as that would mean that the method often yields a transformation with a small translation error. One sees that, for consecutive frames, all of the methods reach nearly 100% *success ratio* at a relatively small λ_t. The conclusion is that while AICK using surf keypoints

[10]AICK was run for 25 iterations with $\alpha = 0.8$ and $R_w = \infty$

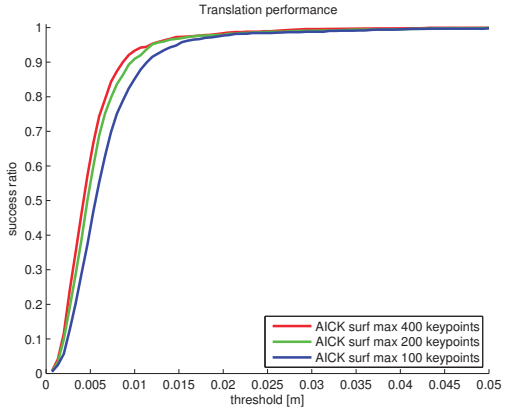

Fig. 5. Effects of limiting the number of surf keypoints used in AICK on the *success ratio* as a function of the threshold on the translation error.

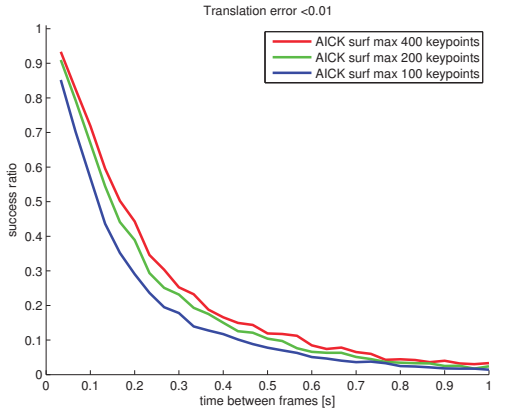

Fig. 6. Effects of limiting the number of surf keypoints used in AICK on the *success ratio* as a function of the time difference between frames with a fixed threshold on the translation error of 0.01 m.

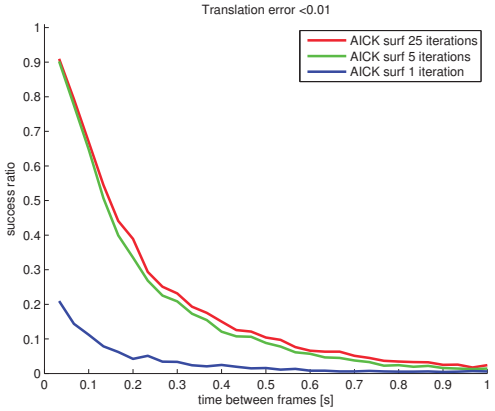

Fig. 7. Effects of limiting the number of iterations using orb keypoints in AICK on the *success ratio* as a function of the time difference between frames with a fixed threshold on the translation error of 0.01 m.

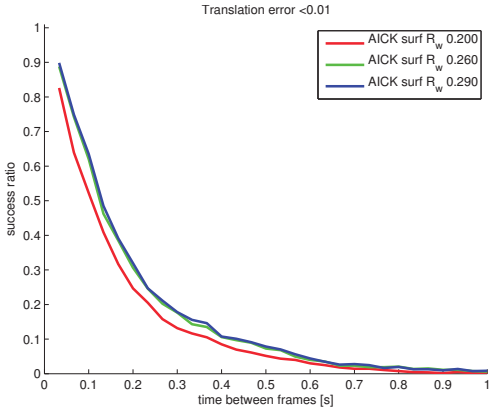

Fig. 8. Effects of changing R_w using surf keypoints in AICK on *success ratio* as a function of the time difference between frames with a fixed threshold on the translation error of 0.01 m.

outperforms the other methods in this test all of the methods are fairly accurate given small displacements of the camera. It is also interesting to note that the difference between the use of surf and orb keypoints is relatively small for AICK.

In order to evaluate robustness to large camera displacements it is informative to see the result on the *success ratio* by using a fixed λ_t and varying the time difference between the frames being matched. This is shown in fig. (4)[10] for a threshold of 0.01 meters. It is clear that the AICK and 3-point RANSAC degrades much slower than GICP and NDT when the camera displacement increases.

A. Convergence and Speed

We can control the runtime to performance trade-off of the algorithm using three main parameters: the number of keypoints used, the number of iterations the algorithm is allowed to run and the threhold R_w. By limiting the maximum number of keypoints used in each frame, the runtime can be significantly reduced. Fig. (5) and fig. (6) indicate that a higher number of keypoints tend to generate better transformations. However, there are clear diminishing returns for using more than 200 surf keypoints. The effect of having many keypoints is also greater when dealing with larger differences in frame capture timestamps. We theorize that the reason is a higher chance of finding stable keypoints if more keypoints are available. Similar investigations show that AICK using ORB keypoints have similar properties, but does however require around 350 keypoints to maintain most of its performance.

In previous tests AICK has been allowed to run for 25 iterations in an attempt to guarantee convergence. The algorithm tends to converge well after the fifth iteration using SURF keypoints given a suitable α value as is apparent in fig. (7). Similar experiments show that AICK using ORB keypoints need about 10 iterations to converge. By looking at the case where the algorithm was run for only one iteration it is clear that the adaptive inclusion of the geometric information greatly reduces the reliance on a good initial keypoint match. The reduction on the number of iterations the algorithm is run significantly cuts down the average runtime per registration.

In order to use the vocabulary of words to speed up our registration we needed to chose parameters for the size of the vocabulary and the R_w. We varied the vocabulary size from 10 to 5,000 and found that above 1,000 there was

Algorithm					*success ratio* for threshold λ_t		
	Keypoints	$\mathbf{R}_w$	**Iterations**	**Avg runtime [s]**	$\lambda_t = 0.0033$	$\lambda_t = 0.01$	$\lambda_t = 0.05$
AICK	on avarage 906 surf keypoints	∞	25	0.180	0.374	0.944	0.993
AICK	on avarage 857 orb keypoints	∞	25	0.135	0.276	0.885	0.999
AICK	max 200 surf keypoints	∞	5	0.00385	0.281	0.902	0.993
AICK	max 350 orb keypoints	∞	10	0.0113	0.209	0.833	0.998
AICK	max 200 surf keypoints	0.26	5	0.000445	0.258	0.888	0.992
AICK	max 350 orb keypoints	0.165	10	0.000717	0.209	0.828	0.995
GICP			25	224	0.070	0.366	0.996
NDT			25	237	0.177	0.706	1
3-point RANSAC			400	4.09	0.255	0.860	0.993

TABLE I

RUNTIME COSTS AND PERFORMANCES FOR THE TESTED ALGORITHMS. R_w IS THE RADIUS AROUND THE KEYPOINT TO FIND MATCHING WORDS.

no improvement in performance which indicated that 1,000 words was a good vocabulary size. In fig. (8) we see how the *success ratio* depends on R_w for surf features.

The effects of these parameter changes together with the *success ratio* of the registration for two consecutive views for different thresholds can be seen in Table I while fig. (9)[11,12] shows the effects on the robustness to large camera displacements. The cost for extracting keypoints used by AICK or 3-point RANSAC is not included in the table. The reason being that in many applications keypoint extratction is only done once per frame whereas frame to frame registration may be run multiple times per frame. For the frames in the test set we found an average of 906 surf keypoints with valid depthdata in an average of 0.12 seconds and an average of 857 ORB keypoints with valid depthdata in an average of 0.011 seconds. It can be seen that the keypoint based methods are much faster than the non-keypoint based methods. Obviously runtime is dependent on implementation but since the keypoint methods deal with significantly less data there are less calculations to be done. By controlling the parameters for the AICK algorithm robustness and precision similar to that of the 3-point RANSAC can be achieved in a fraction of the time.

$R_w = \infty$ indicates not using words at all. Table (I) shows that tuning the parameters of the algorithm can greatly speed up the registration while fig. (9)[11,12] shows that the drop in performance was relatively small.

B. Visual inspection

The results presented above clearly show that AICK outperforms other methods in robustness and precision. In fig. (10) we visualize the results of accumulating transformations estimated by AICK over a sequence of 1000 frames. This is a common and effective way to allow for a qualitative evaluation by visual inspection. Because transformations are added frame by frame, i.e. pure dead-reckoning, errors, especially in orientation, will result in clearly visible distortions. The data is captured at 20 frames per second using [35] with an uncalibrated handheld PrimeSense sensor. To remove the background and avoid displaying noisy data, only data captured close to the sensor is displayed. The absence of distortions during 50s of pure-deadreckoning with the sensor moving in 6D lends credibility to the practical use of the AICK method on real world systems.

[11] AICK orb fast was run 10 iterations with $\alpha = 0.6$, a maximum of 350 orb keypoints and $R_w = 0.165$.

[12] AICK surf fast was run 5 iterations with $\alpha = 0.3$, a maximum of 200 surf keypoints and $R_w = 0.26$.

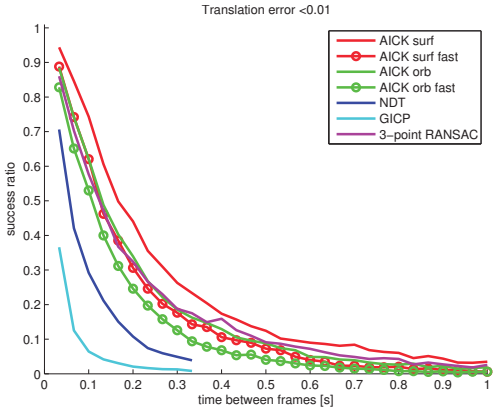

Fig. 9. *success ratio* as a function of the time difference between frames with a fixed threshold on the translation error of 0.01 m.

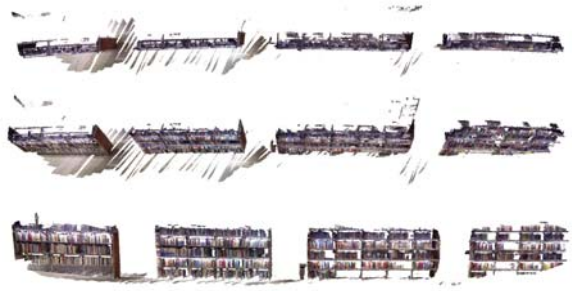

Fig. 10. Rendering of the points given by frame-to-frame transformation estimates when walking next to a series of bookshelves in the KTH library. The data is displayed from three different view points. The bookshelves are lined up in the library and the upper part of the image shows that our method produces results very close to this even using pure dead-reckoning.

VI. SUMMARY AND CONCLUSIONS

We proposed a unified method for transitioning between coarse, appearance-based registration with no initial estimate and fine registration using position-based ICP on distinctive keypoints. In order to verify the performance of our method,

we employed a standard benchmark consisting of a dataset and performance measure [5]. We compared the method to three different high performance registration techniques. In the experiments our method showed a significant improvement of both robustness to larger transformations and precision of the final result which can be attributed to the adaptive distance metric. Furthermore, sub-sampling of the point cloud into a selection of keypoints together with a good search heuristic(close visual words) resulted in an algorithm orders of magnitudes faster than the algorithms used for comparison while providing better performance.

Our results clearly show the feasibility of building real-time systems using visual cues for robust RGB-D data registration. We also believe that the proposed algorithm is a suitable data registration algorithm for real-time, large-scale simultaneous localization and mapping (SLAM) systems.

ACKNOWLEDGMENT

This work was funded by SSF through its Centre for Autonomous Systems and the EU FP7 project STRANDS (600623).

REFERENCES

[1] P. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Trans. on Pattern Analysis and Machine Intel.*, no. 2, pp. 239–256, 1992.

[2] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Proceedings of Robotics: Science and Systems*, (Seattle, USA), June 2009.

[3] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3d-ndt," *Journal of Field Robotics*, vol. 24, pp. 803–827, 2007.

[4] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[5] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[6] B. Zitov and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977 – 1000, 2003.

[7] Z. Zhang, "Iterative point matching for registration of free-form curves," *IRA Rapports de Recherche, Programme 4: Robotique, Image et Vision*, no. 1658, 1992.

[8] Y. Chen. and G. Medioni, "Object modeling by registration of multiple range images," in *Proc. of the 1992 IEEE Intl. Conf. on Robotics and Automation*, pp. 2724–2729, 1992.

[9] P. Biber and W. Strasser, "The normal distributions transform: A new approach to laser scan matching," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2003.

[10] T. Stoyanov, M. Magnusson, and A. J. Lilienthal, "Point Set Registration through Minimization of the L2 Distance between 3D-NDT Models," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 14–19 2012.

[11] L. Montesano, J. Minguez, and L. Montano, "Probabilistic scan matching for motion estimation in unstructured environments," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, 2005.

[12] J. Minguez, L. Montesano, and F. Lamiraux, "Metric-based iterative closest point scan matching for sensor displacement estimation," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 1047–1054, 2006.

[13] H. Men, B. Gebre, and K. Pochiraju, "Color point cloud registration with 4d icp algorithm," in *ICRA*, pp. 1511–1516, IEEE, 2011.

[14] A. Johnson and S. B. Kang, "Registration and integration of textured 3-d data," in *3-D Digital Imaging and Modeling, 1997. Proceedings., International Conference on Recent Advances in*, pp. 234 –241, may 1997.

[15] S. Granger and X. Pennec, "Multi-scale em-icp: A fast and robust approach for surface registration," in *European Conference on Computer Vision, 2002*, pp. 418–432, 2002.

[16] G. Sharp, S. Lee, and D. Wehe, "Icp registration using invariant features," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, pp. 90 –102, jan 2002.

[17] J. Salvi, C. Matabosch, D. Fofi, and J. Forest, "A review of recent range image registration methods with accuracy evaluation," *Image and Vision Computing*, vol. 25, no. 5, pp. 578 – 596, 2007.

[18] R. Newcombe, A. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pp. 127–136, IEEE, 2011.

[19] G. Yang, C. V. Stewart, M. Sofka, and C.-L. Tsai, "Registration of challenging image pairs: Initialization, estimation, and decision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1973–1989, 2007.

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[21] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*, pp. 404–417, Springer, 2006.

[22] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: binary robust independent elementary features," in *Computer Vision–ECCV 2010*, pp. 778–792, Springer, 2010.

[23] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2564–2571, IEEE, 2011.

[24] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006*, pp. 430–443, Springer, 2006.

[25] C. Harris and M. Stephens, "A combined corner and edge detector," in *In Proc. of Fourth Alvey Vision Conference*, pp. 147–151, 1988.

[26] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2548–2555, IEEE, 2011.

[27] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 510–517, IEEE, 2012.

[28] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an rgb-d camera," in *Int. Symposium on Robotics Research (ISRR)*, (Flagstaff, Arizona, USA), Aug. 2011.

[29] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pp. 1470–1477, IEEE, 2003.

[30] M. Cummins and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[31] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, pp. 346–359, June 2008.

[32] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *International Conference on Robotics and Automation*, (Shanghai, China), 2011 2011.

[33] C. Evans, "Notes on the opensurf library," Tech. Rep. CSTR-09-001, University of Bristol, January 2009.

[34] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[35] A. Aydemir, D. Henell, P. Jensfelt, and R. Shilkrot, "Kinect@ home: Crowdsourcing a large 3d dataset of real environments," in *2012 AAAI Spring Symposium Series*, 2012.