DeReEs: Real-Time Registration of RGBD Images Using Image-Based Feature Detection And Robust 3D Correspondence Estimation and Refinement

Sahand Seifi
Memorial University of Newfoundland
sahands[at]mun.ca

Afsaneh Rafighi
Memorial University of Newfoundland
ar7107[at]mun.ca

Dr. Oscar Meruvia-Pastor
Memorial University of Newfoundland
oscar[at]mun.ca

ABSTRACT

We present DeReEs, a real-time RGBD registration algorithm for the scenario where multiple RGBD images of the same scene are obtained from depth-sensing cameras placed at different viewpoints, with partial overlaps between their views. DeReEs (Detection, Rejection and Estimation) is a combination of 2D image-based feature detection algorithms, a RANSAC based false correspondence rejection and a rigid 3D transformation estimation. DeReEs performs global registration not only in real-time, but also supports large transformation distances for both translations and rotations. DeReEs is designed as part of a virtual/augmented reality solution for a remote 3D collaboration system that does not require initial setup and allows users to freely move the cameras during use. We present comparisons of DeReEs with other common registration algorithms. Our results suggest that DeReEs provides better speed and accuracy especially in scenes with partial overlapping.

Categories and Subject Descriptors

I.4 [**Computing methodologies**]: Image processing and computer vision; I.4.5 [**Computing methodologies**]: Image processing and computer visionReconstruction

General Terms

Algorithm, experimentation

Keywords

registration, alignment, 3d, reconstruction, multi-camera, feature-based registration, rgbd

1. INTRODUCTION

The cost effective availability of depth sensing cameras has made it possible to use 3D point clouds in many fields of research and industry. Depth cameras can be used to easily create partial 3D models of a scene or an object of interest. To gather more information from the scene and build a more complete 3D model that covers all sides, the depth cameras can be moved around the scene/object as in 3D scanning applications (such as [13]) or multiple depth cameras might be used simultaneously from different angles as in Virtual and Augmented Reality scenarios [15]. In other scenarios such as indoor robot localization (e.g. [7]), the camera is set up on the robot and changes to the model indicates the movements and the location of the robot.

In all these scenarios, to accumulate or compare point clouds from different frames that are taken from different positions in the same scene, the point cloud coordinates need to be transformed into a unified coordinate system to determine their relevancy with each other. Registration or alignment algorithms receive two point clouds as input and estimate the transformation that transforms the coordinates of one cloud into the other.

There are three main challenges regarding the existing registration algorithms. 1) Initial Pose: some algorithms heavily rely on good initial pose of the input point clouds, or in other words, low transformation distance between the two point clouds. Otherwise, they are prone to local optimums (Figure 1). 2) Speed: less restrictive and more accurate algorithms are not well suited for real-time applications. 3) Extent of overlap: most algorithms are designed with the assumption that the input 3D images share all or most of the scene and no major portion of the scene belongs to only one of the 3D images.

The aim of this research is to design a registration technique that can be used in scenarios where point clouds from 3D frames with *large pose distances* can be aligned together in *real-time*. The motivation behind this research is to create a remote 3D collaboration software where participants' physical models and their surroundings are captured and shared with other participants as virtual or augmented reality using multiple depth sensing cameras at each end. Limiting the number of cameras requires the registration algorithm to perform well with partially overlapping image pairs with large transformation distances. Furthermore, for a smooth user experience the system must not require users to perform fine tuning and alignment for use, while having minimal restrictions over how the cameras are set up. A real-time registration algorithm is required to hold alignment if users decide to move the cameras during communication or if they bump into the cameras, which is foreseeable considering the obstruction of vision by VR/AR wearable glasses or head-mounted gears.

In this work, we propose DeReEs, a simple registration algorithm that can handle large transformation distances both in terms of rotation and translation and places no significant restrictions over the 6 degrees of freedom. DeReEs is a refined version of the work in progress presented in [21]; with improvements in the estimation of the final fine transformation. It performs well with partially overlapping pairs (23% overlap or more) that cause most algorithms fall into local optimums. Our experiments indicate that it performs

IVCNZ '14, November 19 - 21 2014, Hamilton, New Zealand
Copyright 2014 ACM 978-1-4503-3184-5/14/11 $11.00
http://dx.doi.org/10.1145/2683405.2683447 .

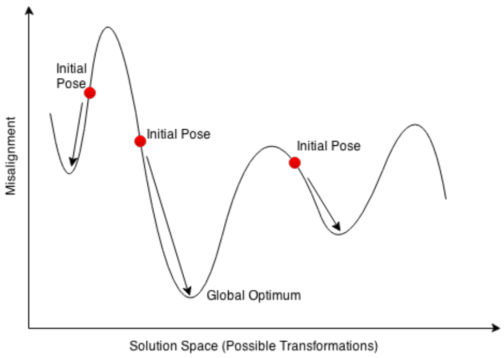


Figure 1: Importance of initial pose in algorithms such as ICP.

faster ($27fps$) and more accurate than existing algorithms.

2. RELATED WORK

2.1 ICP

The ICP (Iterative Closest Point) algorithm originally introduced by Chen and Medioni [9] and Besl and McKay [5] is widely used and is well suited for registration of 3D point clouds with minimal transformation distances; for example in scenarios where the depth camera is capturing a scene with high frame-rates while moving slowly as in hand-held 3D scanners.

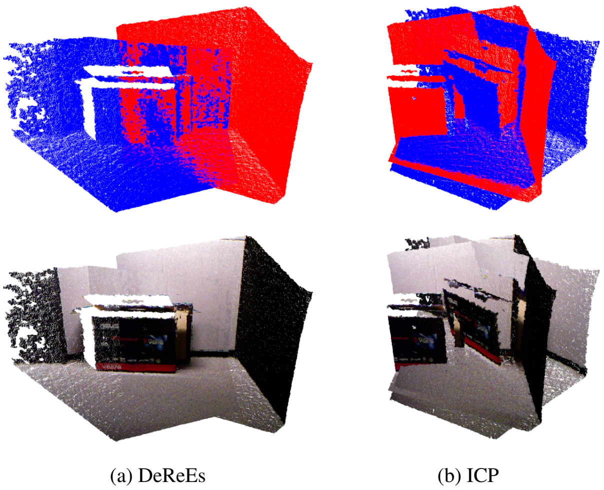
ICP starts from an initial pose, which can be the default pose from the 3D images or a coarse transformation found by another algorithm. Then, a collection of corresponding pairs from both clouds are selected (e.g. by closest distance or normal projection), an error metric is calculated between the corresponding points (e.g. least square [2]) and a transformation that minimizes the error metric is applied. The process continues iteratively until a criteria is met. The criteria is usually set as a number of iterations or when the most recent transformation is negligible. Numerous variants of ICP (such as [22], [14], [10] and surveyed in [20]) are proposed that focus on individual steps of the algorithm.

With any of the ICP variations, the algorithm tends to fall in local optimums if a poor initial pose of the point clouds (large transformation distance) and large amounts of non-overlapping points prevent successful corresponding pair matching. With clouds that have large non-overlapping parts, minimizing the error metric forces the exclusive parts of the clouds to be pushed together. This is illustrated with an example in Figure 2. The correct transformation would appear to have a large error measurement with distance-based metrics since the non-overlapping sections between both point clouds increase the total distance reported. In such scenarios, the ICP algorithm would favour and estimate an incorrect transformation in an attempt to decrease the error metric (visualized in Figure 2b), unlike our proposed solution (Figure 2a).

2.2 3D-NDT

NDT (Normal Distributions Transform) was first used by [6] for 2D registration and was extended to 3D registration by [18]. In his work, Magnusson explains the short-comings of using point clouds: 1) lack of surface characteristics such as orientation or smoothness, 2) when extracted from sensors, point clouds have unnecessaryly large number of points on surfaces close to the sensor and much less information for further surfaces.

Instead, 3D-NDT is designed as a point-to-surface registration algorithm: instead of using a point cloud, the source model is transformed into a smooth surface representation. This representation consists of a set of local probability density functions which describe different sections of the surface.



(a) DeReEs (b) ICP

Figure 2: Registration results of an image pair with large non-overlapping parts with our proposed algorithm and ICP. The color of the points indicate the point cloud they belong to.

This is done by dividing the model into a grid of cells (cube for 3D, square for 2D) and computing the Probability Density Function (PDF) for each cell, which describes the likelihood of a point existing in a certain position in that cell. PDF is a piecewise smooth representation of the surface in each cell that has continuous derivatives, so it also describes the orientation and smoothness of the surface. The algorithm then aims to find a transformation that maximizes the likelihood of the point existing in the surface.

Choosing the right cell size is a challenge in the NDT algorithm. Overly large cell sizes will cause the algorithm to neglect details of the model, since the computed PDF function uses normal distribution and will blur little features. In other words, large cells lead to less detailed registration. Overly small cell sizes on the other hand, will cause the algorithm to fail if the initial pose of the models are not close to the solution, since cells only contribute to the target points that are within their boundary. Magnusson also explains many extensions to the 3D-NDT algorithm that aim to cover its limitations, especially with cell sizes. While 3D-NDT generally performs better than ICP in terms of transformation distance (based on [18]), it is not suited for real-time applications, as registration of each frame pair requires > 1 seconds to complete [17].

2.3 Feature-Based Registration

Feature-based registration on 2D images can be used in most of the commercial depth-sensing cameras as they incorporate a RGB sensor and provide colored point clouds. Scale Invariant Feature Transform (SIFT [16]) or Speeded Up Robust Features (SURF [4]) algorithms are used with the majority of feature based registration techniques to indicate corresponding points based on visual features of the images.

Compared to ICP, the correspondence accuracy is much higher with these methods, but the number of corresponding pairs is very low. There are two main problems with existing feature-based registration methods: 1) a single false feature corresponding pair can

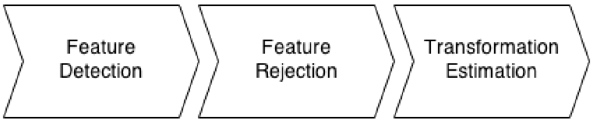


Figure 3: Proposed Registration Pipeline.

create significant errors considering the low number of corresponding pairs, 2) feature-based registration is prone to false feature-pair detection when the scene is dynamic, as a moving object can throw off the registration completely, or when similar objects exist in the scene. Depending on the implementation of the feature detection algorithm, the speed of these techniques might not be suitable for real-time applications.

It is necessary for feature-based registration techniques to incorporate a robust method for rejecting false feature pairs, as seen in RGBD-ICP by Henry et al. [12]. RGBD-ICP uses image-based feature detection algorithms with true pair detection for coarse registration and combines it with the ICP algorithm for further refinement of the transformation. This work successfully addresses the Initial Pose problem with the ICP algorithm, however by using ICP it inherits ICP's weakness against partially overlapping 3D images. In the case of a large non-overlapping set of images, the first stage of the algorithm provides the ICP algorithm with a good estimate of the solution, but ICP tends to move away from this solution by pushing the non-overlapping parts of the images together to minimize the error.

3. DEREES

The pipeline of the registration method we propose is depicted in Figure 3. The input of the pipeline is a pair of 3D images captured from cameras placed at different positions which share a certain portion of the scene and the output is a transformation matrix.

3.1 Detection

2D feature detection algorithms are used to extract features from the RGB images. Multiple algorithms might be executed and results can be aggregated for a potentially more accurate alignment in exchange of speed. Feature descriptors are extracted and features from each image are matched together to derive corresponding feature pairs. The output of this step is two sets of corresponding features (one set for each image) with each feature described by its 2D position in the RGB image.

The GPU implementations of the SURF and ORB [19] feature detection algorithms, as well as the GPU implementation of brute force feature matching have been used in our system.

For the next step of the algorithm, 3D coordinates of the detected feature pairs are required which can be easily extracted for each point based on their corresponding depth values in the depth channel. Considering that the depth map from depth sensing cameras is incomplete compared to the RGB information, a noticeable number of feature pairs are lost at this step. Missing depth information mainly occurs for two reasons:

- Depending on the type of depth camera, depth values may not be present for shiny surfaces, very dark objects or light sources.
- The RGB sensor and the depth sensor are located slightly apart from each other in the depth sensing camera. Hence, parts of the RGB information may not correspond with the depth information.

3.2 Rejection

Not all the pairs from previous step are true corresponding pairs. False corresponding pair refers to the situation when the matching algorithm has matched two unrelated feature points. In this step we aim to find 3 true feature pairs using Random Sample Consensus (RANSAC [11]). This helps us estimate a coarse 3D transformation based on these 3 features and detect all false corresponding pairs based on the coarse transformation.

3 pairs can be randomly selected to be considered as the true pairs; but the reliability of this random selection is determined by the ratio of true and false pairs.

$$\alpha = \frac{TrueCorrespondingFeaturePairs}{AllCorrespondingFeaturePairs} \quad (1)$$

Considering that α is the ratio of true corresponding feature pairs to all pairs, the probability of selecting 3 true pairs from the set with one try is α^3. With scenes where $\alpha = 0.5$ the chances of a correct coarse transformation would be at only 12.5%.

We assume that by repeating random selection for enough times, 3 true feature pairs will be eventually selected with high certainty. While the probability of missing 3 true features with one try is $1 - \alpha^3$, if the selection is repeated n times, the probability that none of the selections consist of 3 true feature pairs (P_{miss}) is $(1 - \alpha^3)^n$. n can be selected in a way that the P_{miss} falls below the desired certainty threshold.

$$P_{miss} = (1 - \alpha^3)^n \quad (2)$$

$$n = \log_{1-\alpha^3} P_{miss} \quad (3)$$

With $\alpha = 0.5$ and 69 repetitions, the probability that no 3 true feature pairs are randomly selected falls below 0.01%.

The 3 true features from any of the "good" random selections can be used to generate the desired coarse transformation based on the transformation between the 3 features and their respective pairs, but a scoring is required to identify the good selection(s) from missed selections.

The correctness of a random selection can be assessed based on the correctness of the transformation it produces. Conventionally, the correctness of a registration transformation is assessed by applying the transformation to the point clouds and then measuring the distance of the two clouds, using metrics such as least square. This is time consuming and inaccurate for the similar reasons mentioned in subsection 2.1: non-overlapping areas of two clouds will cause the distance metric to increase while the clouds might be aligned correctly, since points in exclusive parts of the clouds have no neighbouring points in the other cloud. Conversely, an incorrect alignment that pushes the non-overlapping parts together may produce a better value for the distance metric.

In this work, the proposed metric for evaluating the correctness of the transformation is the number of all features that their distance to their respective pairs is less than a certain distance threshold (d) after applying the transformation. A correct transformation will transform the point cloud in a way that truly corresponding pairs will be placed in close proximity, while false apparent pairs will have noticeable distances. Our metric counts the number of true pairs by taking advantage of this observation.

Note that due to depth noise and inaccuracy in the depth map, camera resolution and inaccuracy of the feature detection algorithm, even true pairs would not be identical in terms of (X, Y, Z) values after the transformation is applied. The need for using a distance threshold in the scoring process arises from this issue. The distance threshold is selected in a way that these inaccuracies along with the inaccuracy of a coarse transformation does not cause true pairs to be rejected (5cm in our experiments).

Unlike true pairs, transformations between most false pairs are arbitrary and it is extremely unlikely that an incorrect transformation results in a majority of false pairs being placed close by. In the case of an incorrect transformation, the 3 feature pairs will have relatively small distances and all true pairs, as well as the majority of false pairs, will have noticeable distances.

The following is the process followed to obtain the coarse transformation:

1. Random selection: 3 feature pairs are randomly selected - $(A1, B1, C1)$ from first view and their respective pairs $(A2, B2, C2)$ from second view.
2. Transformation estimation: transformation T is estimated in way that T transforms $(A1, B1, C1)$ to $(A2, B2, C2)$.
3. Transformation: T is applied to all feature pairs.
4. Scoring: the pairs that their distance is less than the threshold are counted.
5. Repeat steps 1-4 n times.
6. Transformation with the best score is selected as coarse transformation (T_c).

Finally, the coarse transformation T_c is applied to all feature pairs and all the pairs that have a greater distance than the distance threshold d are removed from the set.

3.3 Estimation

The remaining feature pairs are used to estimate a transformation which best transforms the features from one cloud to their corresponding features by minimizing the least squares error between the two sets. Performing this with only the true pairs ensures that false feature pairs do not affect the overall accuracy. Minimizing the least squares is performed by an algorithm based on singular value decomposition (SVD) proposed by Arun et al. [2], with equal weights for all corresponding pairs.

4. EXPERIMENTS

To generate our data-set, a Microsoft^{TM} Xbox 360 Kinect camera is used to capture color+depth images at 640 × 480 resolution. Image pairs are captured in moderately lit indoor environment. The ground truth is estimated by visually aligning each of the image pairs used in the experiments with an accuracy of $1cm$ for translation and 0.5° for rotation. The supplementary material, including the image pairs we used for the experiments are available for the research community at [1].

The workstation used for executing the tests is an Intel Core-i7 960 (3.2Ghz) machine with a NVidia Quadro K5000 GPU, running Ubuntu 12.04 LTS 64-bit. For our experiments GPU implementations of the SURF and ORB feature detection algorithm and Brute Force feature matching have been used from the OpenCV library [8]. The Point Cloud Library has been used for depth-sensing input, visualization tools and implementation of ICP and 3D-NDT for comparison purposes.

The distance threshold d explained in subsection 3.2 is set in a way that true corresponding pairs are not rejected because of a slightly incorrect coarse transformation and depth errors. We have set d to 5cm as it has proven most successful across all data-sets. Unless mentioned explicitly, the number of RANSAC iterations is set to 150.

Method	Success (%)	$Error_t$ (cm)	$Error_r$ (degree)	Speed (ms)
3D-NDT	30	16.12	21.87	21063
ICP	40	18.11	17.36	42.37
RGBD-ICP	NA	12.3	3.3	730
DeReEs	96	1.11	0.21	36.68

Table 1: Comparison of DeReEs, ICP, 3D-NDT and RGBD-ICP in terms of success ratio, transformation error and speed.

Algorithm	Overlap (%)	$Error_t$ (cm)	$Error_r$ (degree)	Success (%)
NDT	83	39.04	17.40	10
ICP	70	62.89	10.12	91
DeReEs	23	1.28	2.68	100

Table 2: Performance of DeReEs, ICP and 3D-NDT under different amounts of overlap between the image pairs of the same scene.

4.1 Comparison

DeReEs is compared to ICP and 3D-NDT using a data-set of 10 RGBD image pairs with low to moderate transformation distances, each being executed 10 times for every algorithm (100 executions in total for each algorithm). Table 1 outlines the success ratio, transformation error and speed of all three algorithms. The RGBD-ICP values in table 1 represent the evaluations reported in the publication ([12]). Due to lack of access to implementations of registration algorithms other than ICP and 3D-NDT, a direct performance comparison was not possible with other algorithms such as [23] and [3].

We defined any transformation estimation with a translation error of $0.5m$ or more for any axes or a rotation error of 30° around any axes as an unsuccessful registration. Transformation error for each transformation is calculated as the sum of absolute translation errors in the 3 axes and the sum of absolute rotation errors around the 3 axes. The errors in the table ($Error_t$ and $Error_r$) represent the average of these errors only for successful registrations, as to prevent unsuccessful registrations from skewing the error measurements.

The speed is the average execution time of all executions. Since the speed of both ICP and 3D-NDT is highly dependent on the size of the point clouds, it is essential to down-sample the $300k$ input of the Kinect camera. The clouds were down-sampled to $6 - 7k$ points for ICP and $2 - 3k$ points for 3D-NDT.

Our algorithm performs at 36.68ms per frame (including 2D feature detection), capable of processing RGBD images at $27fps$. As seen in the table 1, DeReEs outperforms other algorithms in all metrics. It should be noted that it is possible to run both ICP and 3D-NDT faster by a more strict down-sampling, which in turn decreases accuracy. A direct speed comparison with the RGBD-ICP cannot be made due to different settings and configurations, although it is known that the majority of the time that RGBD-ICP uses is dedicated to the ICP algorithm with dense input for transformation refinement ($500ms$). It should be noted that we have tested the RGBD-ICP approach by using ICP in the Estimation step of our algorithm with no noticeable effect on the accuracy, and as earlier described in subsection 2.1, prone to partially overlapping images.

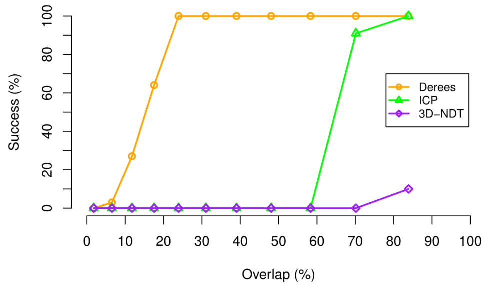


Figure 4: Success rate of DeReEs, ICP and 3D-NDT under different amounts of overlap between the image pairs of the same scene.

4.2 Performance Comparison Under Varying Amounts of Overlap

To evaluate how our algorithm performs with varying degrees of partial overlap between image pairs against other algorithms, we generated a set of overlapping images from a scene. We did this by capturing an initial image from the scene, then rotating the camera by 5 degrees around one axis before taking the next image and so on. The first image is used as the source point cloud, while consequent images are used as target point clouds. Considering Kinect camera's field of view of 57 degrees, the amount of overlap between each image pair is calculated as the ratio of the overlapping portion of the clouds with respect to the whole coverage of the two clouds based on the amount of camera rotation r, where r is the amount of rotation from the initial pose (5° for second image, 10° for third image, etc).

$$OverlappingRatio = \frac{57 - r}{57 + r} \tag{4}$$

Table 2 represents the average of error measurements and the success rate for the minimum amount of overlapping with different algorithms, before the results deteriorate with less overlap. Note that unlike the previous experiment, the error measurements in this experiment are the average for all executions. Figure 4 illustrates the success rate between the three algorithms.

DeReEs outperforms other algorithms by successfully registering image pairs (such as figure 5) with only 23% of overlap, compared to requiring of 70% and 83% overlap for ICP and 3D-NDT. Robustness of our algorithm deteriorates with less overlap as the feature detection and matching algorithms begin to provide a large number of false corresponding pairs. For this experiment, RANSAC iterations has been set to 200, while still providing a real-time solution ($25fps$).

4.3 Robustness Under Poor Image Conditions

Texture-less surfaces and low number of detectable features in the scene greatly affect the performance of the 2D feature detection algorithm. In such conditions, the randomized selection of 3 pairs for coarse registration estimation has a lower chance of success. Hence, the number of iterations used at this step becomes the crucial determinant in successful registration in poor conditions, which in return affects the speed of the algorithm.

We evaluated the accuracy and the speed of our algorithm with different number of RANSAC iterations (from 10 to 200 - See figure 7). We performed this experiment with a feature-less image pair against a feature-rich one (Figure 6). The feature detection algorithm has also been constrained to produce less certain results.

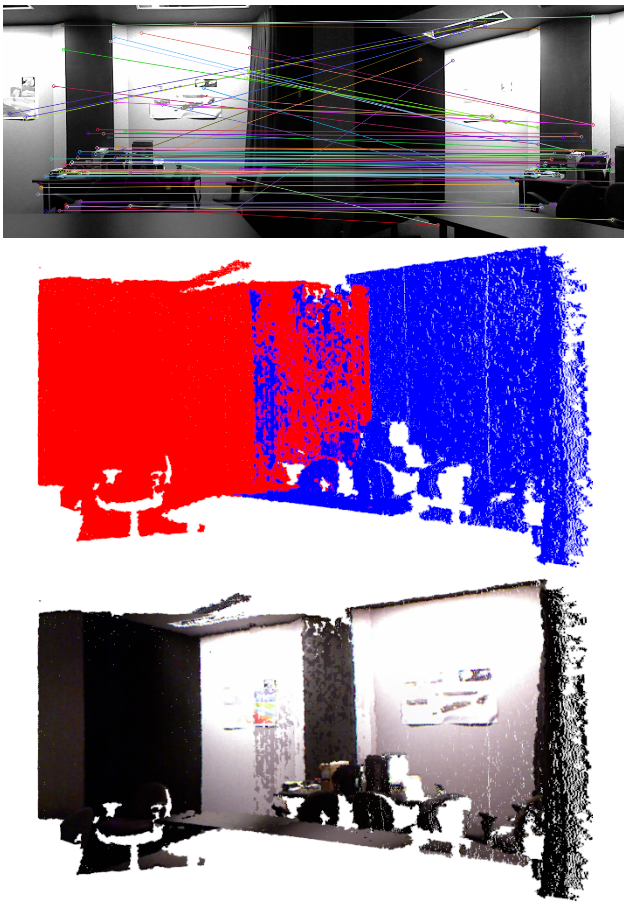
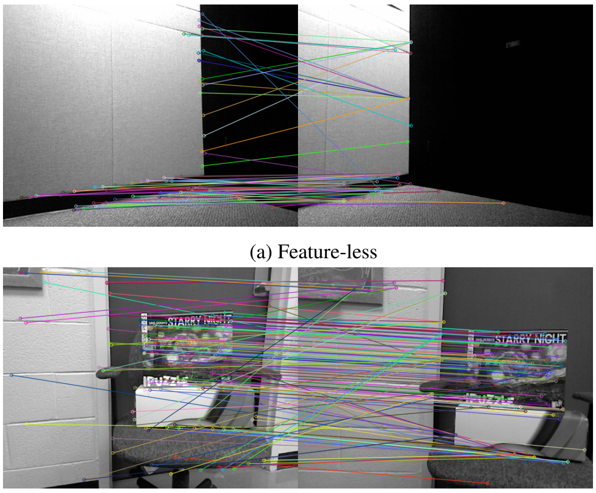


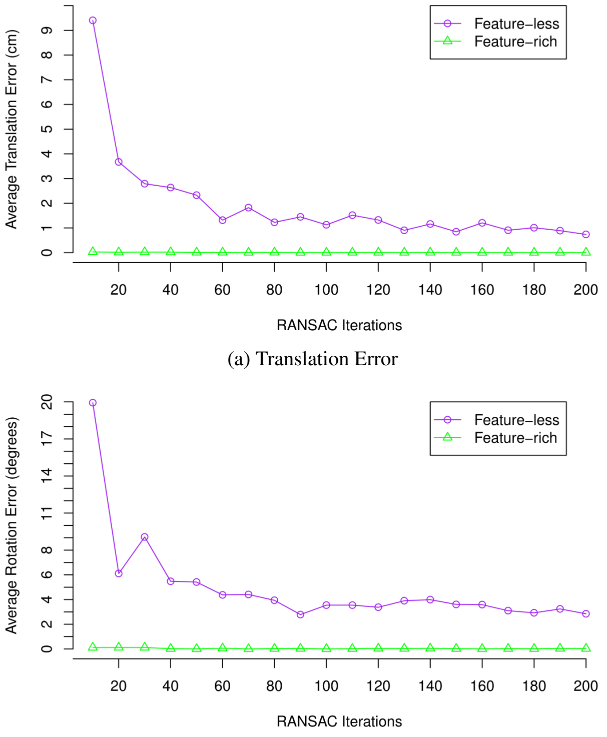
Figure 5: Successful registration of image pairs with only 23 percent overlapping using DeReEs. Top: The individual RGB images from each viewpoint. Lines indicate the feature matchings. Middle: The point clouds for each RGBD image after registration. Bottom: Successfully registered RGBD image pair.



(a) Feature-less

(b) Feature-rich

Figure 6: Comparing feature-less with feature-rich scenes.



(a) Translation Error

(b) Rotation Error

Figure 7: Effects of increasing the number of RANSAC iterations on accuracy of the algorithm, compared between a feature-less and a feature-rich 3D image pair.

Figure 7 illustrates the effect of increasing the number of iterations on transformation errors on each of these pairs. Each data point is the average result of 100 executions.

As seen in Figure 7, increasing the RANSAC iterations has a direct effect on increasing the accuracy of the results which is necessary for maintaining the registration accuracy of the feature-less scenes. With a feature-less scene, our algorithm can outperform other algorithms at 170 iterations, generating $0.90cm$ translation and 3.09° rotation errors while still processing frames at $34.90ms$.

5. CONCLUSION

The algorithm runs in real-time and is not limited to largely overlapping image pairs or a very limited range of translational or rotational distances. Transformation refinement can be further enhanced to achieve optimal solution in complex scenarios. As other visual feature based registration techniques, this algorithm is dependent on RGB information. We believe this algorithm is a suitable solution for real-time scenarios with multiple depth camera settings (e.g. virtual reality collaborative cave), reducing the burden of set up (calibration and/or manual alignment) or misalignment in case of accidental camera displacements; or with single depth cameras in static scenes.

6. REFERENCES

[1] Derees - supplementary materials: Data-sets and ground truth. `http://tiny.cc/derees`. Accessed: 2014-07-09.

[2] K. Arun, T. Huang, and S. Blostein. Least-squares fitting of two 3-d point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-9(5):698–700, Sept 1987.

[3] C. Audras, A. Comport, M. Meilland, and P. Rives. Real-time dense rgb-d localisation and mapping. *Australian Conference on Robotics and Automation*, 2011.

[4] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *In ECCV*, pages 404–417, 2006.

[5] P. Besl and N. D. McKay. A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, Feb 1992.

[6] P. Biber and W. Strasser. The normal distributions transform: a new approach to laser scan matching. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 3, pages 2743–2748 vol.3, Oct 2003.

[7] J. Biswas and M. Veloso. Depth camera based indoor mobile robot localization and navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1697–1702, May 2012.

[8] G. Bradski. Opencv. *Dr. Dobb's Journal of Software Tools*, 2000.

[9] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proceedings. IEEE International Conference on Robotics and Automation*, pages 2724–2729 vol.3, Apr 1991.

[10] S. Druon, M.-J. Aldon, and A. Crosnier. Color constrained icp for registration of large unstructured 3d color data sets. In *Information Acquisition, 2006 IEEE International Conference on*, pages 249–255, Aug 2006.

[11] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[12] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *In the 12th International Symposium on Experimental Robotics (ISER*. Citeseer, 2010.

[13] D. Huber. Automatic 3d modeling using range images obtained from unknown viewpoints. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 153–160, 2001.

[14] T. Jost and H. Hugli. A multi-resolution icp with heuristic closest point search for fast and robust 3d registration of range images. In *Proceedings. Fourth International Conference on 3-D Digital Imaging and Modeling*, pages 427–433, Oct 2003.

[15] G. Kurillo, R. Bajcsy, K. Nahrsted, and O. Kreylos. Immersive 3d environment for remote collaboration and training of physical activities. In *Virtual Reality Conference, 2008. VR '08. IEEE*, pages 269–270, March 2008.

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[17] M. Magnusson. *The Three-Dimensional Normal-Distributions Transform — an Efficient Representation for Registration, Surface Analysis, and Loop Detection*. PhD thesis, ÃŰrebro University, Dec. 2009. ÃŰrebro Studies in Technology 36.

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

[19] 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*, pages 2564–2571, Nov 2011.

[20] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152, 2001.

[21] S. Seifi, A. Rafighi, and O. Meruvia-Pastor. Real-time registration of highly variant color + depth image pairs. In *Poster Proceedings of the Graphics Interface Conference*, 2014.

[22] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, pages 311–318, New York, NY, USA, 1994. ACM.

[23] T. Whelan, H. Johannsson, M. Kaess, J. Leonard, and J. McDonald. Robust real-time visual odometry for dense rgb-d mapping. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5724–5731, May 2013.