Using RGB Information to Improve NDT Distribution Generation and Registration Convergence

James Servos* and Steven L. Waslander†
University of Waterloo, Waterloo, ON, Canada, N2L 3G1

***Abstract*— Unmanned vehicles are becoming an inevitability in our society and with them comes the need for highly robust and accurate algorithms to perform their critical functions, such as localization and mapping. The proliferation of these robots into wide spread use requires a generalized, robust SLAM solution. This paper proposes an improved NDT algorithm, which is capable of performing robust, accurate localization and mapping in an broad spectrum of possible environments and with a multitude of different sensors. The method uses a color greedy cluster approach to cluster points and generate Gaussian distributions and then uses an exhaustive color weighted distribution to distribution cost function to optimize the scan alignment. With the addition of these key features to the NDT framework the method is capable of providing accurate results with minimal computation time. Evaluation is performed on both the Freiburg and Ford datasets to demonstrate a multitude of environments and shows robust registration throughout a wide range of environments and viewpoints.**

I. Introduction

Simultaneous localization and Mapping (SLAM) is a vital part of any functional unmanned system. Unmanned vehicles are being deployed to perform a wide variety of missions in a diverse range of settings. Many of these environments have never been mapped before, whether it be the bottom of the ocean [1] or a disaster zone [2], and require the vehicle to localize and map in an unknown environment.

In recent years SLAM algorithms have progressed significantly, enabling faster more robust localization and more refined mapping. Current state-of-the-art algorithms include both dense camera-based methods, and sparse laser-based methods, both of which have positive and negative aspects which make them suited for different sets of applications.

Dense SLAM methods are ideal for inspection tasks in close-quarters environments. Dense methods are capable of creating high detail maps of small to medium sized areas with great accuracy. Newcombe et al. proposed KinectFusion [3], which uses a Kinect sensor to generate dense 3D maps of small areas. The method was further extended by Whelan et al. [4] to allow larger working volumes and include color. These methods rely on GPU parallelization in order to be able to perform in real time, and can be extremely memory intensive as the mapped region grows. Another method, known as RGBDSLAM [5], has been developed by Endres et al. and uses RGBD images collected from a Kinect or stereo camera to track image features in 3D space and optimize them using a graph SLAM backend. This approach can produce accurate maps but can slow down significantly as the graph grows and can struggle is non-feature-rich environments. Unlike the previous methods, Dense Tracking and Mapping (DTAM) [6], created by Newcombe et al., uses a single monocular camera to generate 3D reconstructions of environments. This method creates accurate reconstructions but can not determine scale without an outside source and is ideal for close up applications. All of these methods require dense close quarters information in order to perform effectively and struggle to handle larger, more homogeneous environments. They generally have significant difficulty in making large quick movements and using long range measurements such as are expected in outdoor operations.

Methods used for sparse SLAM applications are typically suited for medium to large scale environments with possibly large movements between measurements, such as in outdoor navigation. These methods often employ scan registration techniques to solve for the movement between laser scans. The most common scan registration technique is the Iterative Closest Point (ICP) method first introduced by Chen and Medioni [7]. ICP attempts to minimize the nearest neighbour distance between points in two scans. ICP has been improved and expanded upon in many different ways, most notably by Segal et al. who create Generalized ICP (GICP) [8] also know as Plane to Plane ICP. GICP can result in very accurate registration results, however, as it must iterate through every point and therefore can be slow for large scans.

A parallel school of scan registration techniques, introduced by Biber and Strasser [9], is the Normal Distribution Transform (NDT). The NDT method divides the base scan into a grid and calculates a Gaussian distribution from the points in each grid cell. Scans are then aligned by minimizing the point to distribution cost of each point in the input scan to the distribution in the target scan within the corresponding cell. Further expansion by Stoyanov et al. [10] demonstrated that using a distribution to distribution cost function could improve scan registration results over ICP and point to distribution NDT. The NDT method can be tuned for accuracy or speed based on the size of the grid cells used, but the grid representation can be none ideal for some environments, and the cell size must be changed as the environment changes. To remove the need to tune the grid cell size, Magnusson et al. suggest a multi-scale approach [11] which runs multiple NDT optimizations at multiple scales. This approach significantly improves the convergence of the registrations but also significantly increases computation time in the process.

* M.A.Sc. Candidate, Mechanical and Mechatronics Engineering, University of Waterloo; jdservos@uwaterloo.ca

† Assistant Professor, Mechanical and Mechatronics Engineering, University of Waterloo; stevenw@uwaterloo.ca

In general, scan registration methods can be quite time consuming as they require the cost function to be evaluated at every point in a scan or at least a large subset of points or derived distributions in order to obtain accurate results. This can be a problem for vehicles traveling at high speeds or in dynamic environments as the localization and mapping results may be delayed. Also, these methods, as well as many other SLAM methods, can fall victim to geometric degeneracies in spaces with minimal structure in the environment.

Recently, Das et al. [12] showed that by using a greedy clustering approach to generate surface Gaussians, a smooth and continuous NDT cost function could be created and could be evaluated significantly faster than previous methods. This approach however is not ideal in all environments as it requires object to be disjoint in 3D space in order to be accurately clustered. Conversely, Color NDT by Huhle et al. [13], uses color information to generate multiple distributions per cell and match distributions and points based on color weightings. This method helps reduce the effects of geometric degeneracies but can be slow if the grid is made too small, or can be inaccurate if the grid is to large.

This paper proposes the use of a combination of greedy clustering and color weightings to augment the NDT algorithm in order to attain consistent, robust performance in all environments and applications. The proposed method uses color and 3D information in a greedy clustering approach to cluster points and generate the requisite Gaussian distributions used for the optimization phase. The optimization function uses the calculated Gaussians to perform a weighted distribution to distribution cost between all pairs of distributions. The weightings are determined based on a normal distribution of the expected variance in color channels for the given sensor configuration. This approach gives a smooth, continuous cost function which can be optimized efficiently, reduces local minima, and avoids issues with geometric degeneracies when varied color information is present. This method also performs well in both dense, close range situations, as well as large scale sparse outdoor environments.

The proposed method is evaluated using the Freiburg RGBD Dataset [14] as well as the Ford Vision and Lidar Dataset [15]. The Freiburg dataset includes RGBD information collected from a Kinect Sensor in several indoor environments as well as corresponding ground truth information. The indoor environment demonstrates an example of dense SLAM data of a small area with moderate sensor motions. The Ford dataset is generated from a vehicle mounted Velodyne laserscanner and Ladybug omnidirectional camera and demonstrates outdoor, road speed traversal of a typical unmanned vehicle. The Ford dataset evaluates the algorithm in a relatively spares scenario where the sensors are moving at high sped.

II. Normal Distributions Transform

The Normal Distributions Transform is a common form of scan registration used in modern robotics applications. Many authors have proposed improvements to the algorithm but the underlying principles have remained constant. The NDT method attempts to model sets of points as Gaussian distributions and match these distributions either to points or corresponding distributions in the input scan.

The general scan registration problem can be defined as finding the optimal transformation, $T \in \mathbb{SE}(3)$, which best aligns a set of input points, $X = \{x_1, \ldots, x_{N_X}\}$ where $x_i \in \mathbb{R}^3$ for $i \in \{1, .., N_X\}$, with a set of target points, $Y = \{y_1, \ldots, y_{N_Y}\}$ where $y_j \in \mathbb{R}^3$ for $j \in \{1, .., N_Y\}$. The solution is typically computed by optimizing a cost function defined between the scans X and Y.

The original NDT method begins by first dividing the point in the scan into a set of equally sized cells, c_i. For each cell a Gaussian distribution, $G_{c_i} = \mathcal{N}(\mu_{c_i}, \Sigma_{c_i})$, is calculated using the points from the scan which are within the boundaries the cell. At the end of this process the entire scan is converted into a set of Gaussian distributions, $G = \{G_{c_1}, \ldots, G_{c_N}\}$. In the case of distribution to distribution NDT this step must be performed on both the target and input scan to obtain sets of distributions for the input, G^X, and target, G^Y. The total number of distributions, N, is largely dependent on the size of the grid cells used to divide up the scan. Larger grid cells will result in fewer total Gaussians but small details can be lost in the Gaussian model. Small grid cells are capable of maintaining finer detail, however, this will result in a significant increase in the number of Gaussians and a corresponding increase in computation time. The size of the grid cells often has to be set and reset given different environmental conditions.

Next, the NDT cost function used to optimize the registration transform is calculated using the set(s) of Gaussians computed in the previous step. Both point to distribution (P2D) and distribution to distribution (D2D) have similar cost functions. The point to distribution cost function contribution, $J_{P2D}(x) : \mathbb{R}^3 \rightarrow \mathbb{R}$, of an individual point, x, is defined as:

$$J_{P2D}(x) = \exp(-(x - \mu_{c_i})^T {\Sigma_{c_i}}^{-1}(x - \mu_{c_i})) \qquad (1)$$

where x is the point being evaluated and μ_{c_i} and Σ_{c_i} are the mean and covariance of the cell c_i to which x is corresponded. The sum of the individual point costs is then minimized:

$$T^* = \arg\min_T \sum_{i=1}^{N_X} J_{P2D}(Tx_i) \qquad (2)$$

The resulting transform is the optimal relative movement of the sensor frame between the target and input scans that minimizes the cost function. Unfortunately, the overall cost function will be discontinuous. The correspondence of a point x to a specific distribution, c_i, is dependent on the transformation, T. As the transformation changes the points of the input scan change cells relative to the target scan and therefore correspond to a different distribution and cause a discontinuity when this change occurs. These discontinuities

can be problematic for optimization as the derivative is not well defined and convergence can not be guaranteed. The cost function also demonstrates that in the case of geometric degeneracy the solution will not converge as there will be no sensitivity of the cost function to changes in the degenerate transform direction.

III. Color Greedy Cluster

The clustering method used in this paper is based on the Color-Based Segmentation of Point Clouds method proposed by Zhan et al. [16]. The algorithm proceeds in two main stages. First a greedy region growing process is performed to make rough color consistent regions. Second a region merging and refinement stage is performed in order to refine the rough regions into the final segmentation result.

The greedy region growing stage follows a standard greedy clustering approach and uses color information to determine region inclusion. The set of clusters, $\zeta = \{c_1, \ldots, c_{N_c}\}$, where each cluster is a set of points, $c_i = \{p_1, \ldots, p_{N_{c_i}}\}$, is calculated by adding all points to the unlabeled set, $\mathcal{U}$, and incrementally removing a point from $\mathcal{U}$ to begin a new cluster and adding the point to the open set K. The function $\mathcal{K}(p, k)$ uses the kd-tree nearest neighbour algorithm defined in [17] to find the k nearest neighbours of a point p. The algorithm then checks the nearest neighbours of each point in K to determine inclusion in the current cluster, c_i, based on color difference to the initial cluster point. The function, $\delta(x, y) : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$, defines the color difference between point x and y as the Euclidean norm of the difference of the color vectors. If the nearest neighbour point is sufficiently similar, $\delta(x, y) < \kappa_c$ where κ_c is a threshold parameter, then the point is added to K. Once K is empty, the current cluster is added to the cluster set ζ, and a new cluster is started by selecting a new point from $\mathcal{U}$. The process is repeated until all points are labeled. The flow of the algorithms can be seen in Algorithm 1.

The second stage, region merging and refinement, uses the set of clusters, ζ, found in the previous stage and attempts to merge each cluster to its nearest neighbour clusters based on the adjacent cluster being of the same color. If two nearest neighbour clusters are of similar color, they are merged. Unlike the original method proposed in [16], for this work we do not explicitly merge regions below a certain threshold of points with its nearest neighbour. These regions are assumed to be outliers and are simply discarded from the set of clusters. It is important to note that since the computational complexity of calculating the cost function in Equation 5 scales quadratically with the number of Gaussians generated, N_c, it is helpful to limit the maximum number of distributions considered. A simple trimming of the smallest and largest clusters can be performed to stay within a defined maximum. It can be assumed that very large clusters provide minimal information in terms of overall alignment that the smaller clusters don't already capture and that the smallest clusters are most likely noise or areas that will be hard to match between scans.

Algorithm 1 Color Region Growing

```
i ← 0
ζ ← ∅
K ← ∅
U ← P
while U != ∅ do
  K ← p ∈ U
  while K != ∅ do
    i ← i + 1
    p ∈ K
    for q ∈ K(p, k) do
      if δ(p, q) < α and q ∉ K then
        K ← K ∪ {q}
        c_i ← c_i ∪ {q}
        U ← U\q
      end if
    end for
    K ← K\p
  end while
  ζ ← ζ ∪ {c_i}
end while
```

The final stage is to convert the clustered points into a corresponding set of Gaussians, $G = \{g_1, \ldots, g_{N_G}\}$, where each Gaussian maintains the geometric mean, μ^g, as well as a color mean, μ^c, along with the geometric covariance, Σ. Figure 1 shows the clustering results on an indoor office scene. Note that the Gaussians are of various sizes and shapes and accurately represent the underlying structure of the scene with a minimal set of distributions.

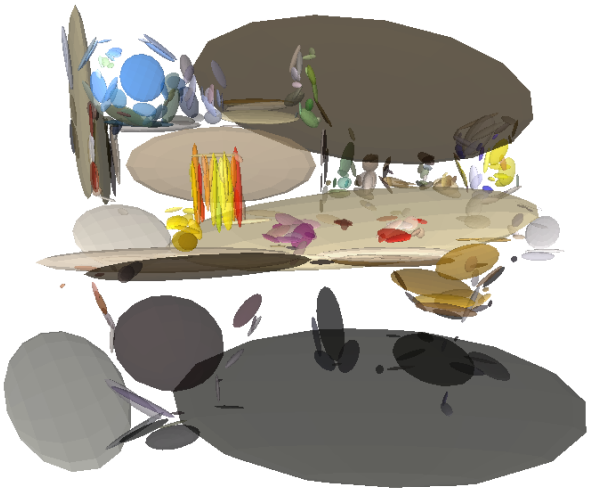
IV. Color Clustered NDT

Using the sets of Gaussian distributions calculated in Section III, for both the input scan, G_i, and the target scan, G_j, the transformation optimization proceeds using a modified cost function. The cost function uses a fully connected approach, where each distribution is evaluated against all other distributions, and incorporates a weighting based on the color difference between the two Gaussians, g_i and g_j, being evaluated. The color weights, λ_{ij}, are calculated given the covariance of the color channels, Λ, defined by the accuracy of the sensor used, and is defined as:

$$\lambda_{ij} = \exp(-\frac{1}{2}(\mu_i^c - \mu_j^c)^T \Lambda^{-1} (\mu_i^c - \mu_j^c)) \tag{3}$$

where $\mu_i^c \in \mathbb{R}^3$ is the mean color of Gaussian g_i. The Gaussians are therefore weighted such that distribution pairs with significant differences in color have a minimal impact on the cost. The final cost evaluation is simply the original distribution to distribution NDT cost weighted by λ_{ij} and calculated between all combinations of distributions. The cost for a single distribution to distribution registration, with a transformed difference of means, $d_{xy} = T\mu_x^g - \mu_y^g$ would be:

(a) Pointcloud of example scene before clustering



(b) Clustering results of the example scene

Fig. 1. Clustering results of an example scene using the Color-Based Segmentation of Point Clouds method and converting point clusters to Gaussians

$$J_{D2D}(g_x, g_y) = \lambda_{xy} \exp(-\frac{1}{2} d_{xy}^T \left[T^T \Sigma_x T + \Sigma_y\right]^{-1} d_{xy}) \tag{4}$$

The minimization of this cost,

$$T^* = \arg\min_T \sum_{i=0}^{N_{G_i}} \sum_{j=0}^{N_{G_j}} J(g_i, g_j) \tag{5}$$

optimizes the alignment of the scans such that distributions of similar color should converge together and result in a more accurate, more robust registration result.

V. Experimental Results

The method proposed in this paper was evaluated based on two datasets, the Freiburg RGBD Dataset and the Ford Vision and Lidar Dataset. Both of these dataset contain color and 3D information as well as ground truth position information. For both datasets the method is evaluated based on the accuracy of the transform compared to the ground truth data as well as qualitatively based on the aggregation of the scans into a consistent final map. The method is evaluated against standard NDT and GICP for both datasets. Finally a comparison of computation time for registering a set of scans is performed.

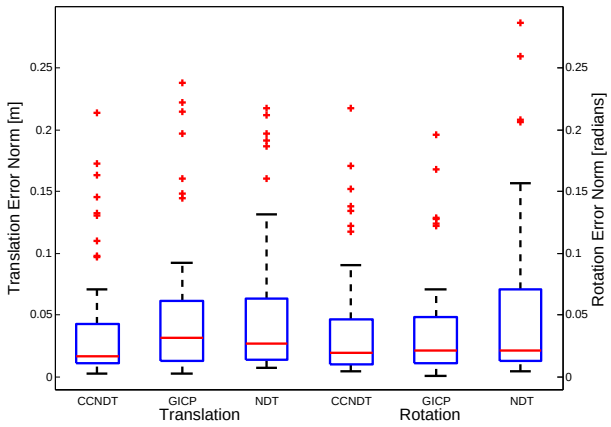


Fig. 2. Freiburg dataset registration transformation accuracy for frame to frame matching for GICP, NDT, and CCNDT, compared to ground truth data.

The evaluation of the method accuracy is performed in a frame to frame arrangement and the computed transformation is compared to the relative motion of the sensor in the ground truth data. The translational and rotational errors are calculated as the Euclidean norms of the respective translational and rotational parameters. The NDT and GICP evaluations are performed with default parameters unless otherwise specified.

A. *Freiburg RGBD Dataset Evaluation*

The Freiburg Dataset was collected using a Kinect RGBD sensor to capture both depth and color images. For this work the depth and color images are combined to create colorized pointclouds which are used with the proposed method. Ground truth information is collected using a MotionAnalysis motion capture system. The Freiburg Dataset consists of dozens of different scenes of indoor environments in a wide range of conditions. For this work the *Long Office Household Scene* scene is used as the benchmark. This scene was selected because it moves through a range of motions and consists of a typical close-quarters indoor scene. This dataset consists of 162 scans from the sensor traversing an approximately 21m path.

Figure 2 shows a box plot of the translational and rotational errors for each frame to frame match. In this case the grid size for NDT was set to 0.1m. This plot shows that Color Clustered NDT has very high accuracy results. The median error in translation for CCNDT is 1.7cm compared to 2.9cm for GICP and 2.7cm for NDT. The rotational errors are only marginally different with CCNDT having 0.019rad and GICP and NDT both with 0.021rad median error. The error plot also shows that the CCNDT has a tighter bounds on

Fig. 3. Aggregate pointcloud map of Freiburg dataset using Color Clustered NDT

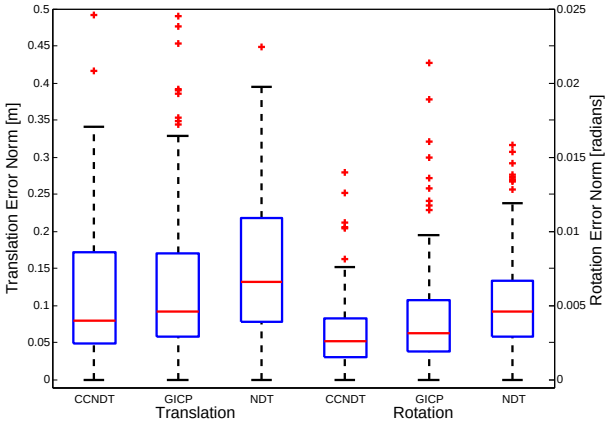


Fig. 4. Ford dataset registration transformation accuracy for frame to frame matching for GICP, NDT, and CCNDT, compared to ground truth data.

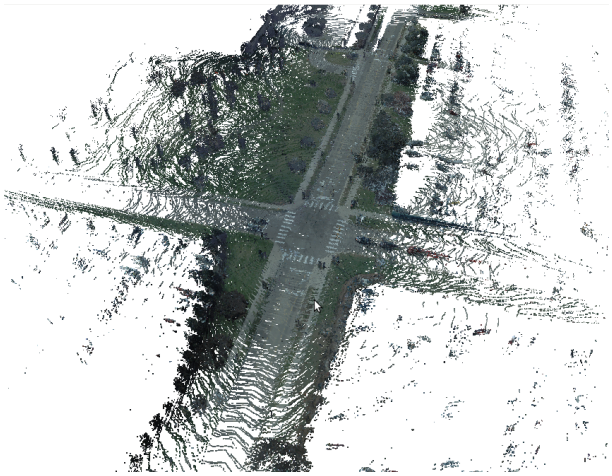


Fig. 5. Aggregate pointcloud map of Ford dataset segment using Color Clustered NDT

its error distribution compared to the other methods leading to more consistent results. Of particular note is that although GICP and NDT are able to perform admirably on this dataset due to a high degree of geometric diversity, in situations of geometric degeneracy, such as a flat wall, both GICP and NDT can fail catastrophically. Overall Color Clustered NDT in the dense indoor environment has highly accurate, robust registration results.

The aggregate map generated using Color Clustered NDT can be seen in Figure 3. The map is seen to be very consistent over the entire dataset.

Since the method proposed in this work is a SLAM front end all of the results represent relative error in frame-to-frame matching. This makes comparison to methods with backend implementations, such as RGBDSLAM, challenging as they will have a clear advantage by using a global optimization. Implementation of this method with a back-end graph optimizer and comparison to a more extensive set of RGBD algorithms is left to future work.

B. Ford Vision and Lidar Dataset Evaluation

The Ford dataset was collected using a Velodyne laser scanner combined with a Ladybug omnidirectional camera. With the accurate extrinsic calibration, the Ladybug camera is used to color the Velodyne point cloud by projecting each 3D point into the camera image. Ground truth information is collected using an Applanix Position and Orientation System. The sensors are mounted to a Ford F-150 truck and data was collected as the truck traversed a stretch of road around the Ford Research Campus. A 200m section of the dataset is used for evaluation purposes. This dataset demonstrates typical conditions of an unmanned vehicle on a roadway.

The box plot shown in Figure 4 shows the overall accuracy of each method in the outdoor environment. For this dataset the grid size for NDT is set to 2m. From this plot it can be seen that GICP and CCNDT have very similar error distributions with CCNDT having only a marginally smaller median value, while both have much lower errors than NDT. The rotational errors in these cases are disproportionately small due to the lack of rotation in the data used. Figure 5 displays a map of aggregate point clouds created using Color Clustered NDT on the Ford dataset. This map shows that in challenging outdoor, sparse environments Color Clustered NDT is still able to generate relatively accurate maps. This is particularly noticeable by the accurate alignment of the road markings visible in the scene.

C. Computation Time

The computation time comparison is performed using the Freiburg dataset, where each scan consists of on average 200,000 points, and using default parameters for each algorithm. The scans are not pre-filtered and are left at full point density for all algorithms. The comparison is performed on a computer with an Intel i7 Quadcore processor and 16GB of ram with each algorithm run in a single thread with no parallelization. Overall computation time for each registration is computed and the results are plotted in Figure 6.

Figure 6 clearly shows that Color Clustered NDT has significantly faster computation times than both standard NDT

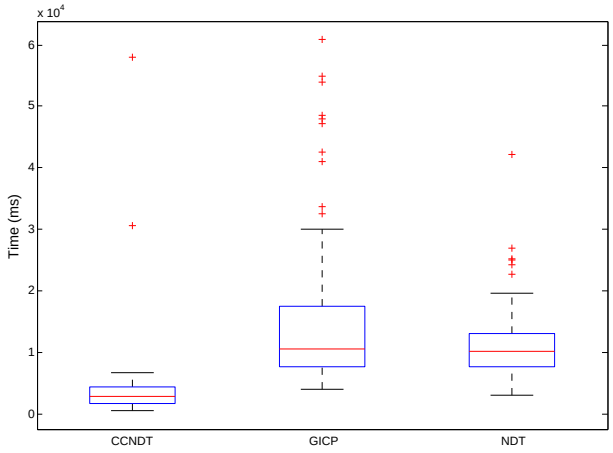


Fig. 6. Computation time comparison between GICP, NDT, and CCNDT, for a single scan registration

and GICP. This is expected because Color Clustered NDT uses significantly fewer clusters to perform the registrations and requires only a single optimization iteration, where as NDT must iterate though each filled cell and GICP through every point.

The boost in computational efficiency of the Color Clustered NDT algorithm would allow it to run on a robot with only a moderately powerful computing platform. This makes it very useful for systems with limited payloads, such as aerial vehicles, or limited power available, such as rovers.

VI. Conclusion

This work presents the Color Clustered NDT method. The presented method incorporates the use of color based greedy clustering to generate the Gaussian distributions and uses color weighted continuous cost function for optimization. This method is shown to have good accuracy in both indoor and outdoor environments and is highly robust to a wide spectrum of environments and conditions. The method proposed is highly versatile and maintains minimal computational complexity. Future work includes evaluation of state-of-the-art clustering approaches to increase clustering performance, implementing the method with a graph-SLAM backend framework, and further evaluation of the method in a range of environments and from wildly varying viewpoints.

References

[1] D. R. Yoerger, M. B. Albert, B. W. Barrie, H. Singh, and R. Bachmayer, "Surveying a subsea lava flow using the autonomous benthic explorer (ABE)," *International Journal of Systems Science*, vol. 29, no. 10, pp. 1031–1044, 1998.

[2] R. Murphy, "Trial by fire [rescue robots]," *Robotics Automation Magazine, IEEE*, vol. 11, no. 3, pp. 50–61, Sept 2004.

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

[4] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended kinectfusion," in *In RGB-D Workshop at Robotics: Science and Systems (RSS)*, 2012.

[5] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 1691–1696.

[6] R. A. Newcombe, S. Lovegrove, and A. Davison, "DTAM: Dense tracking and mapping in real-time," in *IEEE International Conference on Computer Vision (ICCV)*, Nov 2011, pp. 2320–2327.

[7] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *International Conference on Robotics and Automation (ICRA)*, vol. 3. IEEE, Apr 1991, pp. 2724–2729.

[8] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *2009 Robotics: Science and Systems (RSS)*, June 2009.

[9] P. Biber and W. Strasser, "The normal distributions transform: a new approach to laser scan matching," in *International Conference on Robotics and Automation (ICRA)*, vol. 3. IEEE, Oct. 2003, pp. 2743–2748.

[10] T. Stoyanov, M. Magnusson, and A. Lilienthal, "Point set registration through minimization of the L2 distance between 3D-NDT models," in *International Conference on Robotics and Automation (ICRA)*. St. Paul, MN, USA: IEEE, May 2012, pp. 5196–5201.

[11] M. Magnusson, T. Duckett, and A. J. Lilienthal, "Scan registration for autonomous mining vehicles using 3D-NDT," *Journal of Field Robotics*, vol. 24, no. 10, pp. 803–827, Oct 24 2007.

[12] A. Das, J. Servos, and S. Waslander, "3D scan registration using the normal distributions transform with ground segmentation and point cloud clustering," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 2207–2212.

[13] B. Huhle, M. Magnusson, W. Straßer, and A. J. Lilienthal, "Registration of colored 3D point clouds with a kernel-based extension to the normal distributions transform," in *International Conference on Robotics and Automation*. IEEE, 2008, pp. 4025–4030.

[14] 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.

[15] G. Pandey, J. R. McBride, and R. M. Eustice, "Ford campus vision and lidar data set," *International Journal of Robotics Research*, vol. 30, no. 13, pp. 1543–1552, November 2011.

[16] Q. Zhana, Y. Liang, and Y. Xiaoa, "Color-based segmentation of point clouds," in *ISPRS Workshop on Laserscanning (IAPRS)*, vol. XXXVIII, 2009.

[17] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Application (VISSAPP)*. INSTICC Press, 2009, pp. 331–340.