

Cluster-based Scan Matching for Robust Motion Estimation and Loop Closing

Antoni Burguera¹

Abstract—This paper presents a robust approach to estimate the relative motion between couples of range scans called CSoG. The algorithm first searches prominent structural features in one of the scans by means of a clustering algorithm. Thus, no assumptions about the environment are made. Afterwards, it projects the other scan into the detected feature set and uses a score function to evaluate the projection. By optimizing the score function the motion between the two scans is obtained.

Our approach is compared to two well known scan matchers using real data from three different sensors: a terrestrial sonar, a terrestrial laser and an underwater sonar. Results show a significant improvement of CSoG with respect to the other algorithms in the case of medium and large motions between the scans. Accordingly, CSoG is a good choice to perform dead reckoning from range data and to close large loops in SLAM.

I. INTRODUCTION

Localization is one of the most fundamental problems in mobile robotics since almost any task carried out by a robot depends on accurate pose estimates. Even though the use of external beacons can help to solve the problem, they are not always available. Consequently, most of the existing solutions to localization focus on the use of sensors that do not require artificial placement of external devices.

The most commonly used exteroceptive sensors to perform localization are cameras [1] and range finders [2]. The use of cameras has increased in the last years as they provide a richer representation of the environment than range finders. However, the same reason that makes cameras interesting sensors to localize a mobile robot also plays against them. Since cameras provide more information than range finders, more computational resources are needed to properly process the images. Thus, even though the computational power has drastically increased in few years, low cost robots such as autonomous vacuum cleaners can benefit from the use of range finders.

Moreover, cameras are still not the sensor of choice in underwater robotics. In spite of the remarkable results obtained in underwater visual localization, mapping and *Simultaneous Localization and Mapping* (SLAM) [3], most underwater vehicles include sonars as primary sensors [4]. Since sound propagates better than light in marine environments, sonar sensors have significantly larger perception ranges than cameras in that medium.

Independently of the environment being underwater or not, most of the existing range sensors have one feature

that makes them particularly appealing to perform localization: they directly provide geometric information about the environment. Typical sensors such as *Laser Range Finders* (LRF), *Acoustic Range Finders* (ARF) or underwater *Imaging Sonars* (IS), among many others, provide specific coordinates of the obstacles surrounding the robot that can be used to perform localization either directly or with small pre-processing [5].

At the core of every localization system there is an odometer, which is in charge of estimating the motion between couples of consecutively gathered sensor readings. By accumulating these motion estimates, a rough guess of the robot trajectory can be obtained and subsequently refined by means of, for example, SLAM. The measurements provided by range sensors are usually grouped to build the so called *scans*. Thus, the most common approach to odometry using range sensors is known as *scan matching* [6].

Most of the existing scan matchers work at a single reading level. That is, they estimate the robot motion by assuming that individual range readings within each scan are independent among them, thus neglecting the structure of the environment. This is usually seen as an advantage, since trying to fit the scans into certain model reduces the autonomy of the vehicle. For example, trying to model the scans based on straight lines confines the robot operation to man-made structured scenarios.

In this paper we explore a new way to perform scan matching, called *Cluster-Based Sum of Gaussians* (CSoG) taking into account the structure of the environment without compromising the robot autonomy. Our proposal is not hard-code a specific environment structure but to let the system to dynamically build the best representation of the environment. This is achieved by dividing the scan in its most prominent components using a clustering algorithm. Afterwards, the problem of scan matching can be defined as a problem of global optimization, which is also beneficial since it reduces the well known problem of local minima.

II. RELATED WORK

A. Scan matching

Given two scans $S_A = \{p_i = [px_i, py_i]^T, 0 \leq i < M\}$ and $S_B = \{q_j = [qx_j, qy_j]^T, 0 \leq j < N\}$ gathered at the coordinate frames A and B respectively, the goal of a scan matcher is to estimate the motion $X_B^A = [x_B^A, y_B^A, \theta_B^A]^T$ from frame A to frame B. This motion is computed as the one that maximizes the overlap between S_A and $S_B' = X_B^A \oplus S_B = \{q'_j = [qx'_j, qy'_j]^T, 0 \leq j < N\}$, where \oplus denotes the composition transformation. If S_A and S_B have

*This work is partially supported by Ministry of Economy and Competitiveness under contract DPI2017-86372-C3-3-R (AEI,FEDER,UE).

¹Antoni Burguera is with Departament de Matemàtiques i Informàtica, Universitat de les Illes Balears, Carretera de Valldemossa Km 7.5 - 07122 Palma, Spain antoni.burguera@uib.es

been consecutively gathered, scan matching behaves as an odometer. If the scan matcher can properly deal with non consecutively gathered scans, which means large motions between S_A and S_B among others, it can also be used to close loops in SLAM [7].

Scan matching algorithms are usually classified in two rough categories based on their similarity to the well known *Iterative Closest Point* (ICP) [8]. In this way, ICP-based algorithms are those that iterate three steps until convergence. During the first step, the points in S_B are transformed according to an estimate of X_B^A . This means that an initial guess for X_B^A is required. During the second step, each point in the transformed S_B is associated to the closest $p_i \in S_A$. Finally, the third step is in charge of finding the X_B^A that minimizes the sum of squared distances between associated points. Since the associations may not be correct, the algorithm iterates under the assumption that each new set of associations will be better than the previous one.

ICP-based algorithms mainly differ in the distance criteria used during the second and the third step. For example, ICP uses euclidean distance, *Iterative Dual Correspondence* (IDC) [8] takes advantage of polar distances, PLICP [9] makes use of point to line distance and *Sonar Probabilistic Iterative Correspondence* (spIC) [10] relies on statistical compatibility by means of the Mahalanobis distance. The main problem of all these algorithms is that they can easily fall in local minima mainly because the function they have to optimize in the third step changes at every iteration.

The second of the above-mentioned categories refers to those algorithms that do not follow the ICP algorithmic structure. Most of these algorithms build a function that represents S_A and then evaluates each point of S_B on that function. A score function that aggregates these evaluations is then defined and optimized to find X_B^A . Overall, algorithms within this category differ on how they build up the function for S_A . For example, the *Normal Distributions Transform* (NDT) [11] builds a grid, computes the mean and the standard deviation of the $p_i \in S_A$ that lie within each cell and then models each grid cell as a Gaussian, thus defining a piecewise function. Other approaches, such as the *Likelihood Field with Sum of Gaussians* (LFSOG) [12], focus on reducing the problems of the discontinuities among grid cells at the cost of larger computation times.

Our proposal, which lies in this second group, focuses on the way in which the function that models S_A is constructed so that it dynamically takes into account the environment structure by means of clustering techniques.

B. Clustering

Clustering is the process of grouping data samples so that the similarity of the samples is maximum within groups and minimum among them. This is often achieved by defining a distance criteria between samples and also a method to compute a representative, called centroid, of each group. For example, K-Means [13] typically makes use of euclidean distances and centroids are usually defined as the mean of the samples belonging to each group. A variant of this

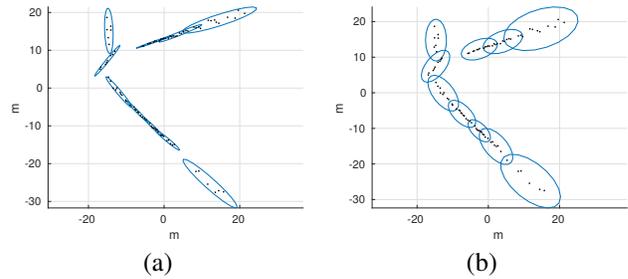


Fig. 1. Example of clusters. (a) Using covariance. (b) Using modified covariance.

algorithm is K-Medoids which basically changes the centroid computation to guarantee that it is a data sample itself. Also, K-Medoids has shown to exhibit better convergence rate than K-Means.

Even though there exist approaches to clustering radically different to K-Means and K-Medoids, such as hierarchical clustering, the need to define a distance able to compare data samples is a constant. Properly defining that distance is crucial, since it constrains the shapes that clusters can have.

Our proposal is to apply a clustering algorithm to find groups of readings in S_A . Since each group will contain readings that are similar between them but dissimilar to other groups, each group will actually represent a salient structural feature of the environment. Thus, a scan matcher could take profit of that structure. However, contrarily to other approaches [14], no assumption about the structure is hard coded: groups are dynamically built according to the structure of each specific scan. Moreover, this also leads to a dimensionality reduction that will significantly reduce the execution time of the scan matcher.

In our case, each data sample has two dimensions and represents a point of an actual object around the robot. Previous studies have shown that Mahalanobis distance offers a fair trade-off between the ability to properly model these kind of spatial measurements [10] and the required computation speed. That is why the clustering approach that will be used in this study is K-Medoids with Mahalanobis distance. Accordingly, our proposal differs from previous approaches to cluster-based scan matching [13] in several points. First, we use K-Medoids since it has interesting properties and faster convergence rate. Second, our metrics are based on Mahalanobis, as it better models the scans than euclidean metrics. Third, our proposal optimizes a unique, global, function thus reducing the local minima problem.

III. THE CLUSTER-BASED SUM OF GAUSSIANS

Our proposal operates in three steps. First, K-medoids with Mahalanobis distance is applied to S_A to find the groups of readings that better capture the structure of that scan. Second, a score function projects S_B into the detected groups and evaluates the goodness of the projection. Finally, the motion that maximizes that score function is searched. That motion is, precisely, X_B^A . Next, these three steps are described.

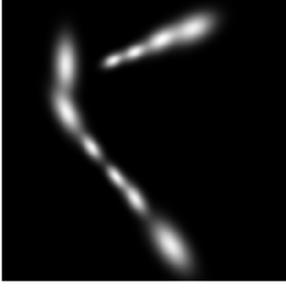


Fig. 2. Example of scan model.

A. Clustering

Let $G_A = \{(\mu_i, \Sigma_i), 0 \leq i < N_G\}$ denote the result of applying K-Medoids to S_A , where μ_i and Σ_i are the mean and the covariance, respectively, of the points assigned to cluster i . Due to the spatial distribution of the scan points, it can be expected that some Σ_i are near singular (Figure 1-a) thus leading to further numerical problems. To avoid this problem, we impose an additional constraint to Σ_i : if the lowest eigenvalue is smaller than one fourth of the largest one, we force it to that value. The result is exemplified in Figure 1-b.

B. The score function

Our proposal is to model S_A using the set of Gaussians defined by the means and the covariances in G_A . In this way, we can represent S_A as a sum of Gaussians. Accordingly, we define the function $f(p)$ as follows:

$$f(p) = \sum_{(\mu_i, \Sigma_i) \in G_A} e^{-(p-\mu_i)^T \Sigma_i^{-1} (p-\mu_i)} \quad (1)$$

where p is a point in the space. The normalization factor has been removed to improve the efficiency since it is not necessary in this case. Figure 2 shows an example of how the previous Equation evaluates for one specific scan. As it can be observed, larger values spread through the regions where the original measurements were located (Figure 1).

We can now define a score function that evaluates the goodness of a motion X from the reference frame of S_A to the reference frame of S_B . Since $f(p)$ provides larger values for points that are close to the actual obstacles, our proposal is to transform the points in S_B by means of X , evaluate each of these projected points by means of Equation 1 and sum all these evaluations. In this way, the better is X , the larger the obtained value will be. That is, the proposed score function $s(X)$ is as follows:

$$s(X) = \sum_{q_j \in S_B} f(X \oplus q_j) \quad (2)$$

C. Optimization

Since the score function in Equation 2 has been defined so that the better the X the larger the output, the goal now is to find the X that maximizes the score. Just to ease

further numeric methods, let us transform this problem into a minimization one by changing the sign of the score function. That is, the motion X_B^A from S_A to S_B can be computed as:

$$\overline{X}_B^A = \arg \min_X \left(- \sum_{q_j \in S_B} f(X \oplus q_j) \right) \quad (3)$$

Our proposal to perform this optimization is to use a trust-region algorithm. To this end, the gradient and the Hessian matrix of the score function to be minimized are required. Since the score function is defined as a sum of exponentials, both the gradient vector and the Hessian matrix can be computed by summing the gradient vectors and the Hessian matrices of the addends respectively. Following explanation focuses on one addend involving (μ_i, Σ_i) and q_j . To ease notation, let us define $\alpha = X \oplus q_j - \mu_i$ so that one addend of the score function can be written as:

$$g(X) = - \exp(-\alpha^T \Sigma_i^{-1} \alpha) \quad (4)$$

One addend of the gradient vector is as follows:

$$\nabla g(X) = \frac{\partial g}{\partial X} = 2 \exp(-\alpha^T \Sigma_i^{-1} \alpha) \alpha^T \Sigma_i^{-1} \frac{\partial \alpha}{\partial X} \quad (5)$$

where

$$\frac{\partial \alpha}{\partial X} = \begin{bmatrix} 1 & 0 & -qx_j \sin \theta_B^A - qy_j \cos \theta_B^A \\ 0 & 1 & qx_j \cos \theta_B^A - qy_j \sin \theta_B^A \end{bmatrix} \quad (6)$$

Note that there is no need to compute the third row since points do not have a θ component. As for the Hessian matrix, one addend is as follows:

$$H = \begin{bmatrix} H_{00} & H_{01} & H_{02} \\ H_{10} & H_{11} & H_{12} \\ H_{20} & H_{21} & H_{22} \end{bmatrix} \quad (7)$$

where

$$\begin{aligned} H_{rc} &= \frac{\partial^2 g}{\partial x_r \partial x_c} = 2 \exp(-\alpha^T \Sigma_i^{-1} \alpha) \cdot \\ &\cdot \left[-2\alpha^T \Sigma_i^{-1} \frac{\partial \alpha}{\partial x_c} \alpha^T \Sigma_i^{-1} \frac{\partial \alpha}{\partial x_r} + \right. \\ &\left. + \left(\frac{\partial \alpha}{\partial x_c} \right)^T \Sigma_i^{-1} \frac{\partial \alpha}{\partial x_r} + \alpha^T \Sigma_i^{-1} \frac{\partial^2 \alpha}{\partial x_r \partial x_c} \right] \quad (8) \end{aligned}$$

The notation x_i in this Equation refers to each of the components of the transformation X , so that x_0 and x_1 denote the x and y displacements respectively and x_2 denotes the rotation angle. Accordingly, $\frac{\partial \alpha}{\partial x_i}$ corresponds to the i-th column of the Jacobian matrix shown in Equation 6. As for the second partial derivatives of α , it is easy to see that they are zero except for H_{22} . That is:

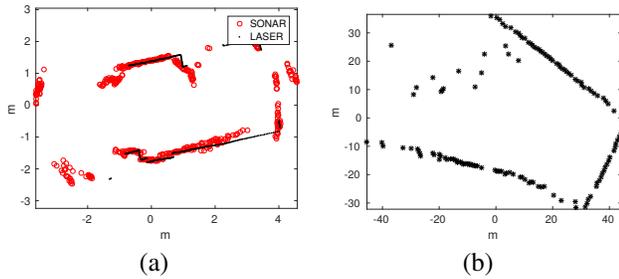


Fig. 3. Examples of scans used in the experiments. (a) Terrestrial sonar and laser. (b) Underwater sonar.

$$\frac{\partial^2 \alpha}{\partial x_r \partial x_c} = \begin{cases} \begin{bmatrix} -qx_j \cos \theta_B^A + qy_j \sin \theta_B^A \\ -qx_j \sin \theta_B^A - qy_j \cos \theta_B^A \end{bmatrix} & r = c = 2 \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{otherwise} \end{cases} \quad (9)$$

It is important to emphasize that, being $g(x)$ of class C^1 (i.e. continuously differentiable), the Hessian matrix will be symmetric. This fact can be used to slightly reduce the computation time.

IV. EXPERIMENTAL RESULTS

Data from three very different range finders has been used to perform the experiments. The first tested sensor suite is a set of 16 terrestrial Polaroid ARF. The data using these sensors was gathered in indoor and semi-outdoor areas in our university using a Pioneer 3-DX robot. The angular uncertainty of these sensors ($\simeq 30^\circ$) combined with the existing multi-path reflections was responsible for a large amount of wrong measurements. To alleviate this problem, readings farther than 5 m have been discarded. Also, because of their low firing rate, measurements had to be grouped along small trajectories to build the scans. In this way we obtained a total of 473 sonar scans involving a mean of 409 readings per scan.

The second tested sensor is a Hokuyo URG-04LX LRF with a maximum range of 5 meters. The sensor was attached to the same robot used to gather sonar data, so that both sensors captured the same environment. Among the gathered laser scans, we selected those that were obtained exactly when the sonar grouping method built one sonar scan. In this way we have a one to one correspondence between the sonar and the laser scans. That is, we also have 473 laser scans. The average number of readings per scan is 378 in this case and their quality is extremely high when compared to the ARF. Figure 3-a shows one scan obtained with the LRF and the corresponding scan gathered with the ARF.

The third tested sensor is an underwater IS. In particular, we have used the data from a *Mechanically Scanned Imaging Sonar* (MSIS) obtained by [15] in an abandoned marina in the Costa Brava (Spain). The MSIS data has been pre-processed in two ways. On the one hand, range readings have been extracted using the technique described in [10]. On the other hand, similarly to the terrestrial sonar, readings

have been grouped along small trajectories to build scans. In this case we have 218 scans with an average of 101 readings per scan. The maximum sensor range in this case is 50 m. Figure 3-b show one of the scans obtained with this sensor.

In order to evaluate our approach, we will compare it to the most representative scan matchers in the two categories mentioned in Section II-A: ICP and NDT, which also are the most widely used approaches to scan matching nowadays. The maximum number of iterations has been set to 1000 for all three algorithms. This means that ICP will re-establish correspondences a maximum of 1000 times and that NDT and CSoG numerical optimization methods will perform a maximum of 1000 iterations. The clustering step in our proposal uses a fixed number of 10 and 20 clusters with underwater and terrestrial data respectively. As for NDT, we used the improved method that builds four overlapping grids and a cell size of 1 m.

For each sensor type (terrestrial ARF, terrestrial LRF and underwater sonar) we proceed as follows. First, each scan is split into two different scans. In this way, we simulate having couples of scans gathered at the same exact position. Thus, the motion between the two scans in each couple is perfectly known to be $[0, 0, 0]^T$. Then, a random roto-translation $X = [x; y; \theta]^T$ is applied to one of the two scans to represent the misalignment between the two scans.

We have defined five levels of misalignment between scans. In level 1, x and y are chosen according to a uniform distribution between -0.01 and 0.01 times the maximum sensor range. The orientation θ comes from a uniform distribution between -5 degrees and 5 degrees. These values increase with the misalignment level, lying between -0.05 and 0.05 times the maximum sensor range in x and y and between -25 degrees and 25 degrees in θ in level 5. All the tested methods are fed with these scans and their output recorded. The process has been repeated 100 times per scan and misalignment level, which means a total of 582000 executions per tested algorithm.

The output of each scan matcher has been classified as true positive (the algorithm converged to a correct motion estimate), true negative (the algorithm did not converge and the estimate at the last iteration was wrong), false positive (the algorithm converged to a wrong motion estimate) or false negative (the algorithm did not converge but the estimate at the last iteration was correct). To do this classification, an estimate is considered correct if its error is below 10% of the maximum sensor range in X and Y and below 10 degrees in orientation.

The results for each scan matcher and dataset are shown in Figure 4. Overall it can be observed that, even though ICP reaches almost 100% of true positives for small misalignments between the scans, its accuracy decreases fast with the misalignment level. The NDT behaves similarly to ICP with slightly better accuracies for largely misaligned scans.

Our proposal, the CSoG, exhibits a slightly lower accuracy for levels 1 and 2 but clearly surpasses all the other methods with all the sensors from misalignment level 3 onward. For example, whilst ICP and NDT have a 42.5% and a 45.6%

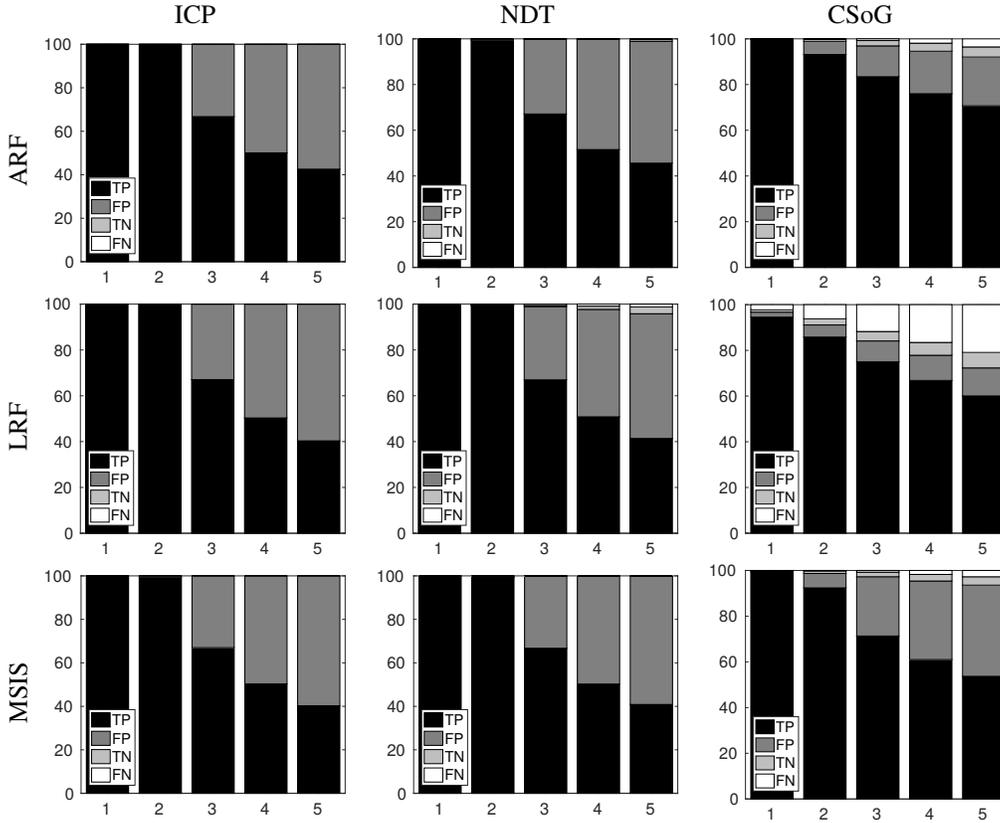


Fig. 4. Comparison in terms of true and false positives and negatives. X axes correspond to the five misalignment levels and Y axes are percentages.

of true positives with misalignment level 5 using terrestrial ARF, the CSoG reaches the 70.7%. A similar behaviour can be observed with the other sensors. Whereas ICP and NDT provide a 40.3% and a 41.4% of true positives in LRF level 5, CSoG reaches a 60.1%. The 40.2% and the 40.9% obtained by ICP and NDT with MSIS data in level 5 is also clearly surpassed by the 53.7% of CSoG.

The results with the LRF are particularly interesting. In this case, CSoG shows significant amounts of false negatives: 11.7%, 16.5% and 20.9% in levels 3, 4 and 5. Since these situations correspond to the cases in which CSoG did not achieve convergence but the estimation in the last iteration was correct, this suggests that improving the convergence criteria or using a better optimization algorithm could significantly improve the CSoG performance.

To provide a clearer understanding of the algorithm's behaviour in front of the misalignments, Figure 5 shows the mean errors of the true positive estimates also as a function of the misalignment level. In particular, the distance error, defined as the euclidean distance from the provided estimate to the ground truth, and the orientation error, defined as the absolute difference between the estimated angle and the ground truth angle, are shown.

Overall, the distance errors are similar between the three algorithms in terrestrial environments but significantly worse for CSoG when using underwater sonar data. As for the orientation error, CSoG provides clearly better results than the other approaches both in the terrestrial and the underwater

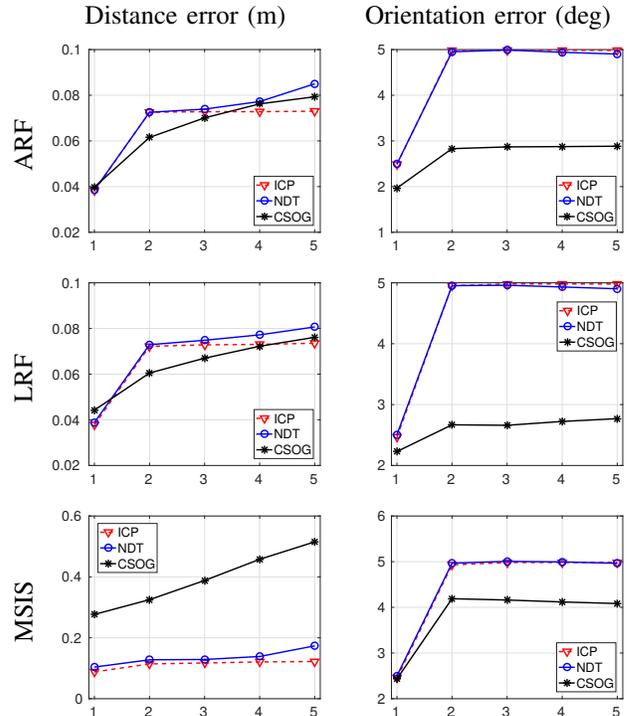


Fig. 5. Comparison in terms of distance and orientation errors. X axes correspond to the five misalignment levels and Y axes are the errors. The distance error is expressed in meters and the orientation error in degrees.

scenarios.

These results lead to some interesting conclusions. First, the ICP and probably ICP-based approaches in general, are the algorithms of choice in front of small misalignments. In these cases, the established correspondences are good enough to guarantee convergence to an accurate solution. Thus, ICP-based algorithms are particularly good to provide odometric estimates. However, the fast drop in performance with the misalignment level makes ICP-based approaches less interesting when it comes to loop closing in SLAM. To the contrary, CSoG performs significantly better than ICP and NDT in presence of largely misaligned scans. Thus, it is well suited to close loops in SLAM, where larger motion errors between non consecutively gathered scans can be expected.

Second, even though there are some differences depending on the sensor being ARF, LRF or MSIS, the relative behaviour between algorithms remain similar independently of the dataset: ICP is the best for small misalignments and CSoG surpasses the NDT for largely misaligned scans.

Taking these facts into account, a future line of research arises: by combining CSoG and an ICP-based approach, accurate results in front of very high noises could be achieved. For example, an initial estimate could be obtained by means of CSoG and used to feed an ICP-based algorithm. In this way, the robustness of CSoG in front of noise could also lead to high accuracy thanks to ICP-based approaches.

V. CONCLUSION

In this paper we have presented a robust approach to estimate the relative motion between couples of range scans called CSoG. The approach operates in two steps. First, it searches prominent structural features in one of the scans. This search is performed by means of a clustering algorithm and, thus, no assumptions about the environment are made. Second, it projects the other scan into the detected feature set and evaluates the goodness of the projection by means of a score function. By optimizing that score function, the motion between the two scans is obtained.

The proposal has been compared to well known representatives of different scan matching approaches. On the one hand, the ICP, that establishes point to point correspondences between scans. On the other hand, the NDT, which also defines a score function similarly to CSoG. All the algorithms have been tested with three different range sensors: a terrestrial sonar, a terrestrial laser and an underwater sonar.

Results have shown that, even though ICP surpasses CSoG in front of low initial errors, its results are not good when the motion between the scans is large. In these cases, NDT and CSoG exhibit better results, being those of CSoG the best ones. Thus, CSoG is a promising algorithm to perform robust loop closing in SLAM.

Future work goes now in two ways. On the one hand, a method to dynamically select the optimal number of clusters would be interesting. Though using a fixed number worked well in the experiments, dynamically tuning it could benefit our approach. On the other hand, we feel that combining

the robustness of CSoG with the accuracy of ICP-based approaches in low noise conditions could lead to extremely accurate loop closing. That is, using CSoG to reduce the error in the initial estimate and then refining it with an ICP-based algorithm could clearly benefit SLAM based on range readings.

Additionally, some other clustering algorithms could be tested. Some initial experiments with an improved K-Means based on Mahalanobis distance have already been performed, exhibiting promising results. Also, hierarchical clustering approaches would make it possible to represent a single scan with different granularities thus allowing to refine the motion estimate depending on the available computational resources.

REFERENCES

- [1] R. Mur-Artal, J. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [2] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 1271–1278.
- [3] A. Kim and R. M. Eustice, "Real-time visual SLAM for autonomous underwater hull inspection using visual saliency," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 719–733, 2013.
- [4] J. Li, M. Kaess, R. M. Eustice, and M. Johnson-Roberson, "Pose-Graph SLAM Using Forward-Looking Sonar," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2330–2337, 2018.
- [5] A. Burguera, "A novel approach to register sonar data for underwater robot localization," in *Intelligent Systems Conference*, 2017, pp. 1034–1043.
- [6] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg, "Globally consistent 3D mapping with scan matching," *Robotics and Autonomous Systems*, vol. 56, no. 2, pp. 130–142, 2008.
- [7] J. Li, H. Zhan, B. M. Chen, I. Reid, and G. H. Lee, "Deep learning for 2D scan matching and loop closure," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, 2017, pp. 763–768.
- [8] F. Lu and E. Milios, "Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 18, no. 3, pp. 249–275, 1997.
- [9] A. Censi, "An ICP variant using a point-to-line metric," in *IEEE International Conference on Robotics and Automation*, 2008, pp. 19–25.
- [10] A. Burguera, Y. González, and G. Oliver, "The UspIC: Performing scan matching localization using an Imaging Sonar," *Sensors*, vol. 12, no. 6, pp. 7855–7885, 2012.
- [11] P. Biber and W. Strasser, "The normal distributions transform: a new approach to laser scan matching," in *IEEE International Conference on Intelligent Robots and Systems*, 2003, pp. 2743–2748.
- [12] A. Burguera, Y. González, and G. Oliver, "On the use of likelihood fields to perform sonar scan matching localization," *Autonomous Robots*, vol. 26, no. 4, pp. 203–222, 2009.
- [13] A. Das and S. L. Waslander, "Scan registration with multi-scale k-means normal distributions transform," in *IEEE International Conference on Intelligent Robots and Systems*, 2012, pp. 2705–2710.
- [14] H. Cho, E. K. Kim, and S. Kim, "Indoor SLAM application using geometric and ICP matching methods based on line features," *Robotics and Autonomous Systems*, vol. 100, pp. 206–224, 2018.
- [15] D. Ribas, P. Ridaio, J. Neira, and J. D. Tardós, "SLAM using an imaging sonar for partially structured underwater environments," in *IEEE International Conference on Intelligent Robots and Systems*, 2006, pp. 5040–5045.