Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 32, Num. 1 (2025) 236-242

RESEARCH ARTICLE

A Mini-map Alignment Approach using Landmark Observation Information Intrinsic to the Feature-based Visual SLAM Pipeline

Uma Abordagem de Alinhamento de Minimapas usando Informações de Observação de Pontos Intrínsecas aos Sistemas de SLAM Visual Baseado em Características

Felipe Pires Saraiva[1]*, Gustavo Teodoro Laureano[1], Thiago Henrique de Oliveira[1]

Abstract: This work proposes a map alignment approach based on a simple modification of the Iterative Closest Point algorithm to use point confidences based on metrics usually available in feature-based visual SLAM pipelines. In the context of a hierarchical map composed of mini-maps, aligning these mini-maps is an important task to allow metric information to be related between them. This research enumerates three possible SLAM metrics that could be used for representing landmark confidence, and investigate the potential of using these metrics to improve the ICP algorithm. The experiments show evidence that the usage of the confidence metrics might help to improve the convergence of ICP with only a small modification to the data association step.
Keywords: Map alignment — ICP — Visual SLAM

Resumo: Este trabalho propõe uma abordagem de alinhamento de mapas com base em uma simples modificação do algoritmo Iterative Closest Point para usar confianças de ponto baseadas em métricas geralmente disponíveis em sistemas de SLAM Visual baseados em características. No contexto de um mapa hierárquico composto de minimapas, alinhar esses minimapas é uma tarefa importante para permitir a relação da informação métrica entre eles. Essa pesquisa enumera três possíveis métricas do SLAM que podem ser usadas para representar a confiança de um ponto de referência do mapa, e investiga o potencial do uso dessas métricas para melhorar o algoritmo ICP. Os experimentos mostram evidências de que o uso das métricas de confiança pode ajudar a melhorar a convergência do ICP com apenas uma pequena modificação na etapa de associação de dados.
Palavras-Chave: Alinhamento de mapas — ICP — SLAM Visual

[1] *Instituto de Informática, Universidade Federal de Goiás (UFG), Brazil*
***Corresponding author**: felipepsaraiva@discente.ufg.br
DOI: http://dx.doi.org/10.22456/2175-2745.143437 • **Received:** 22/10/2024 • **Accepted:** 03/01/2025

1. Introduction

Visual SLAM (Simultaneous Localization and Mapping) enables robots to map an unknown environment while simultaneously estimate its pose in the map, all based primarily on visual information [1]. In this context, hierarchical or hybrid maps, composed of a high-level global map and multiple local mini-maps that cover only a small region of the environment, may be used to represent large-scale environments efficiently while maintaining detailed representation and processing [2].

In the hierarchical mapping and localization approach, metric information is spread across multiple mini-maps, so estimating the transformation between mini-maps is a crucial task, as it allows establishing the relative transformations and metric relationships between them. One possible approach to estimating this transformation in the context of feature-based visual SLAM is through the registration of the mini-maps' point clouds [3].

Iterative Closest Point (ICP) [4] is a widely used algorithm for registering two point clouds, and it is known to work fine when properly initialized, but it originally does not consider the uncertainty associated with the point cloud data, and can be sensitive to noise, outliers, and the initial guess of the transformation [5]. Feature-based visual SLAM pipelines model the base entities of the SLAM problem in similar ways [6], and this produce information, intrinsic to the process, that may potentially represent the confidence of a landmark tracked in the environment and might be used to improve the registration process.

The goal of this research is to enumerate possible SLAM

metrics that could be used for representing landmark confidence, and investigate the potential of using these metrics in the ICP algorithm. By incorporating this additional information to ICP, this work aims to improve the robustness of mini-map alignment. Experiments show evidence that the usage of the confidence metrics might help to improve the convergence of ICP.

The mini-map alignment case is the main motivation of this research, but any scenario of the feature-based visual SLAM where two sub-maps must be aligned can benefit from the findings. Other cases where estimating the transformation between two sub-maps may be useful is the loop-closure procedure, where the transformation between the two visited regions is used to globally correct the map [7], and multi-robot mapping, where each robot builds a local map that are later aligned and fused [3].

2. Related Work

The basic concept of the iterative closest point (ICP) algorithm was first proposed by Besl and McKay [4] and, due to its simplicity and modularity, has since been adopted in a number of research areas, from robotics to 3D reconstruction. Rusinkiewicz and Levoy [8] explored a range of ICP variants that seek to improve an aspect of the original algorithm, while classified them based on which stage of the ICP they affected.

In 2003, Bosse et al. [9] explored the map matching problem in a hybrid metric/topological approach to SLAM to achieve efficient mapping of large-scale environments. The authors found that map alignment is infeasible using the nearest-neighbor matching strategy in situations where the transformation uncertainty is very large. They conducted the experiments considering map uncertainties for data association, suggesting the need to aggregate more information in the ICP algorithm for the context considered in the work.

Ballesta et al. [3] studied the landmark-based map alignment problem for multi-robot SLAM. They compared four alignment methods, including the basic ICP and a proposed modification to improve the initial estimation of the transformation parameters, and applied them to maps built using a SLAM pipeline, showing successful alignments. Zhe Ji et al. [10] presents a variation of the ICP algorithm incorporating the visual ORB feature of the points.

Pomerleau et al. [5] presented a detailed review of point cloud registration algorithms for the mobile robotics domain, focusing on the classical ICP algorithm. At the end of their work, the authors present a table with common challenges of ICP applications and a set of guidelines for each stage. The usage of weights is cited as possible improvements for some challenges, including sensor noise. Bai [11], on the other hand, focused their work on applications of ICP in the context of SLAM, presenting common use cases and other possible approaches.

With the popularity of deep learning, a number of methods based on this technique were also applied on registration problems. Huang et al. [12] surveyed optimization-based and deep learning point cloud registration methods, also summarizing the connections between these approaches. They showed that deep learning methods still does not guarantee accuracy, but the combination with ICP still requires high computation time.

3. Methodology

A basic implementation of a feature-based visual SLAM pipeline keeps track of three base entities: frames, features and landmarks [7]. Each image captured by the camera is a frame, where a number of 2D features are observed. A landmark is a 3D point of interest of the environment that is observed as features in one or more frames. These entities may be modeled differently between each SLAM method, but the essence of the information will generally be available. Since each landmark is associated to a 3D point in the environment, a way of representing the map is a sparse point cloud (PC), and with these base entities and its relationships, it is possible to extract information that could potentially indicate the confidence of each estimated point.

One first option to estimate the confidence of a landmark is the uncertainty of its 3D position. In this case, the covariance matrix of the position is used as the estimate of the uncertainty, and it can be estimated using a probabilistic approach to update the position and its covariance with each new observation. The covariance could also be estimated during a local or global optimization that takes into account the landmark position as optimization variables. Estimating the covariance is an extra step that can be computationally intensive, since it may involve the inversion of highly-dimensional matrices for big estimation problems, as is usually the case for SLAM implementations [13].

The confidence weight is defined to be a scalar, and since a bigger covariance indicates more uncertainty, the confidence weight for the covariance w_c is defined as

$$w_{ci} = \frac{1}{1 + ||Cov_i||} \tag{1}$$

where w_{ci} is the confidence weight of the point i, and $||Cov_i||$ is the L2 norm of the covariance matrix of the same point.

When using a probabilistic approach, more observations and measurements of the same landmark usually lead to greater certainty about its position. By this intuition, the number of observations of each landmark may not represent the uncertainty directly, but may indicate which points would have less uncertainty. In this sense, the confidence weight w_{ni} is defined as the number of observations of the landmark i:

$$w_{ni} = \text{number of observations of landmark } i \tag{2}$$

Lastly, the distance of the landmark 3D point from the camera observing it also influences in the point uncertainty, because when the distance increases, the pixel accuracy decreases and so does the position accuracy [14]. So a third metric of confidence could be the mean distance between the landmark position and the position of all frames that observed

it. In this case, a bigger distance would indicate lower confidence, so the confidence weight for the mean distance w_{di} for the landmark i is defined as

$$w_{di} = \frac{1}{1+\overline{D_i}} \tag{3}$$

$$\overline{D_i} = \frac{1}{N}\sum_{j=1}^{N} |L_i - F_j| \tag{4}$$

where $\overline{D_i}$ is the mean distance of observations, L_i is the position of the landmark, F_j is the position of the frame that did the j-th observation of the landmark, $|L_i - F_j|$ is the Euclidean distance between both positions, and N is the number of observations.

When the same scene is observed twice in the same or multiple SLAM sessions, there are two maps with probably different landmarks but representing the same set of objects. Using the representation of point clouds (PCs) commented earlier, it is possible to estimate the transformation (translation and orientation) between both maps by aligning a source PC with a target PC through the Iterative Closest Point (ICP) algorithm [4], that is a simple, generic and effective algorithm for point cloud registration.

The ICP algorithm performs a simple loop: assume correspondences between the two PCs, estimate the approximate transformation that minimizes the distance between the correspondences, transform the source PC taking it closer to the target PC, iterate. In its most basic form, the correspondence for each point in the source PC is assumed to be the closest point in the target PC, hence the name Iterative *Closest Point*. This is the base implementation used for performance comparison in this work.

After the PCs are aligned, a score can be computed as a measure of the alignment success. The score, as implemented in the Point Cloud Library (PCL) [15], is defined as the mean squared distance between points in the transformed source PC and the closest points in the target PC, formalized as

$$Score = \frac{1}{N}\sum_{i=0}^{N} |S_i - T_i| \tag{5}$$

where N is the number of points in the source PC, S_i is the i-th point of the source PC, T_i is the closest point to S_i from the target PC, and $|S_i - T_i|$ is the Euclidean distance between them. Perfectly aligned PCs would produce a score of 0, bigger scores usually indicate worse alignment, and the weighting strategies are not applied to score computation.

The proposed approach to take into account the confidence weights of the points is a small change in the correspondence estimation step, while the rest of the algorithm is not changed. Instead of finding only the closest point, the method searches for k closest points, also known as nearest neighbors, and use for correspondence the one with the highest weight among them. Considering the set of the k closest points, w_i the confidence of the ith-point, with $1 \leq i \leq k$, the matched point is selected as following equation:

$$Matched\ Closest\ Point = \underset{i}{\operatorname{argmax}}\ w_i \tag{6}$$

The goal of this modification is to keep the simplicity and efficiency of the original ICP, but improving the data association to hopefully avoid points with high uncertainty that could worsen the transformation estimate.

4. Experiments and Results

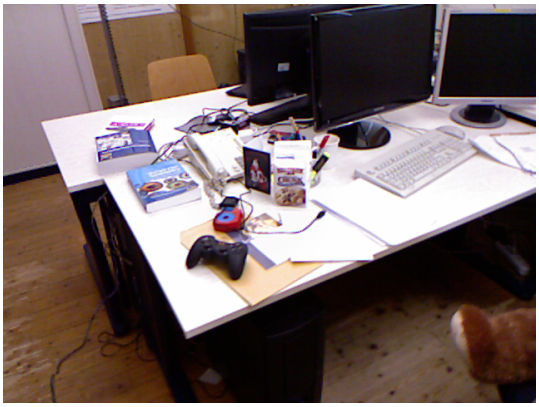
The approach described in Section 3 is not specific to a single implementation of a feature-based visual SLAM pipeline. This work uses a variation of the stereo Visual Odometry detailed in Gao et al. [7], adapted for RGB-D cameras, as the pipeline for building the map. This pipeline is the collection of the essential building blocks of a SLAM system, without even including a loop-closing module. The goal of using this implementation was to attest the contribution of the proposal with a basic implementation of SLAM, without more robust methods to handle drift and alignment issues. Even simple pipelines provide the information used in the approach, so state-of-the-art full Visual SLAM systems like ORBSLAM2 [16] also keep track of such information.

Likewise, the modification proposed for the ICP algorithm is small and impacts only the data association step. For the following experiments, a custom correspondence estimation strategy was implemented in the Point Cloud Library [15], since it provides a modular ICP implementation, but could be easily integrated into other ICP implementations. The code for the ICP module will be available later as supplementary material.

The goal of the experiments is to investigate whether the weights proposed in the last section have the potential to improve the convergence of the ICP method in scenarios close to the one discussed in Section 1, where mini-maps are constructed sequentially and aligned, to build a high-level map with topological information between mini-maps. In this scenario, it is expected that the odometry drift will include a translation error mainly in the XY plane, and an orientation error mainly in the yaw axis, since the camera typically does not have a large movement in the Z axis nor a large rotation around the *roll* and *pitch* axes.

To be able to construct the maps independently, the sequences *fr1/desk* and *fr1/desk2* from the TUM RGB-D dataset [17] were used, which are composed of different camera trajectories over the same office desks. Figure 1 shows one image of each sequence observing the same set of objects from different poses. The map file contains frame poses, landmark positions, covariance matrix, number of observations, and the 2D image coordinates of each observation. For all experiments, the maximum number of ICP iterations was 150.

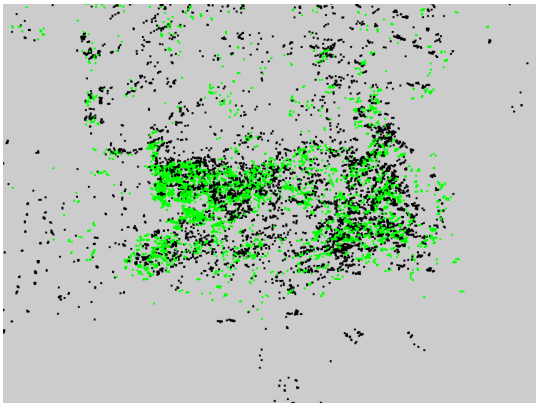
The Figure 2a illustrates an example of different point clouds that are aligned with a score of 0.0034319. Since they



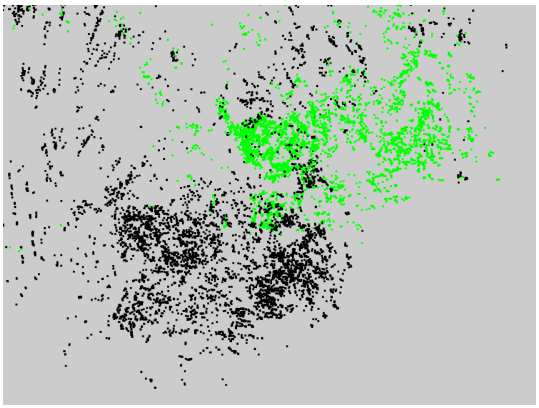
(a) Image 448 of *fr1/desk* sequence.

(b) Image 90 of *fr1/desk2* sequence.

Figure 1. Examples of images from the two sequences of the TUM RGB-D dataset used to build the maps.



(a) Case where the ICP converged with score 0.0034319.



(b) Case where the ICP, with score 0.0167398, failed to align the point clouds.

Figure 2. Examples of alignment for a pair of map point clouds. Each point is an environment landmark detected by the SLAM system. In black is the fixed target point cloud, and in green is the source point cloud that is transformed by the ICP.

are different, the score will difficultly reach *zero*. As shown later in the experiments, in all cases where the alignment is satisfactory, the score is below 0.0037 for this map pair. On the other hand, the Figure 2b shows an example of score 0.0167398 where the ICP could not converge and, therefore, the point clouds remained misaligned. In both examples, the point clouds are the same, but the initial transformation is different.

4.1 Neighborhood size *k*

An important parameter of the alignment approach is the number of close points k that will be considered when searching for the best weight. So for the first experiment the transformation between the two point clouds is fixed while varying k.

Table 1 shows the score for each weighting strategy for a number of k values. It is observed that the base ICP always produces the same score, since it does not use k, and for $k = 1$ all strategies perform the same as the base ICP, since the closest point is always chosen. Starting from $k = 10$, the increase of k also increases the score, which could indicate that a big neighborhood might enhance the chance of finding an outlier correspondence. Between all tested k values, around $k = 5$ seems to yield the best results for the proposed weighing strategies, so for the next experiments this parameter is fixed at the value 5.

4.2 Yaw

With the k parameter defined, the following experiments will focus on analyzing the influence of the initial transformation between the two point clouds on the ICP convergence. At first, the impact of a transformation with only yaw rotation is studied. The range of yaw values starts at 0 degrees, and is incremented by 10 until 70 degrees.

Figure 3 shows the violin plot for the scores obtained for all yaw angles in the experiment. It is observed that the base ICP method rapidly loses convergence with rotation, while the weighted strategies still converge. Only for 70 degrees that the strategies based on the covariance and the number of observations loses convergence.

The mean distance weighting strategy has a unique be-

Table 1. Alignment score obtained by each weighting strategy for various neighborhood sizes k. Each score column represents a correspondence strategy, where **Cov.** is the covariance, **# Obs.** is the number of observations and **Mean Dist.** is the mean distance. In bold is the lowest score for each strategy (column).

k	Score ($\times 10^{-3}$)			
	Base	**Cov.**	**# Obs.**	**Mean Dist.**
1	5.34268	5.34268	5.34268	5.34268
3	5.34268	4.58499	3.3126	**3.31155**
5	5.34268	3.28737	**3.29407**	3.54915
7	5.34268	**3.28614**	3.30512	3.6283
10	5.34268	14.0339	3.33651	4.00365
15	5.34268	14.0491	3.39744	5.54156
20	5.34268	14.0212	3.42316	5.62062

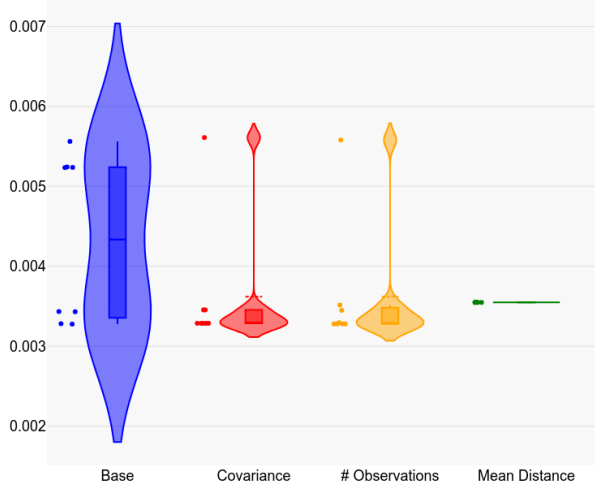


Figure 3. Violin plot of the scores obtained in the experiment varying the Yaw orientation between the two point clouds. The continuous line inside the box is the median value, and the dashed line is the average. On the left of each box, the points represent the individual score values.

havior that is observed in all the following experiments. It appears to be robust to this range of yaw rotation, but it always converges to a score slightly bigger than the scores observed in the other strategies. This may be because the mean distance strategy is prioritizing a better alignment of points close to the camera trajectory at the expense of leaving distant points misaligned, which might be even desired depending on the application.

4.3 XY

The next experiment is intended to investigate the influence that the weights have when the two point clouds are translated in the XY plane, so all other transformation degrees were fixed. Since the direction of translation have great influence on the overlapping region of the point clouds, the experiment was divided in quadrants: XY, $(-X)Y$, $(-X)(-Y)$ and $X(-Y)$. For each quadrant, both X and Y were translated the same amount, by the values $0.25m$ and $0.5m$, and then increased until at least one ICP strategy stopped aligning successfully, when getting a score above 0.0037, as discussed earlier.

Figure 4 shows the violin plot for the scores obtained in the experiment for all quadrants. It is observed that all strategies perform relatively well under small translations, but in three of the four quadrants, the base ICP method was the first to stop converging, while the other strategies were still converging. Also, all weighted strategies showed less variance than the base ICP.

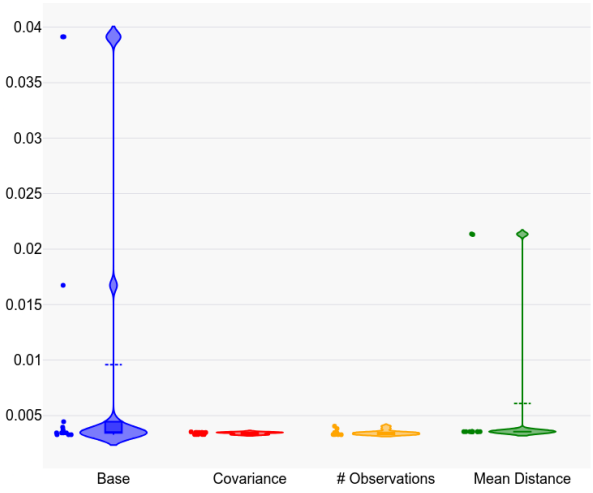


Figure 4. Violin plot of the scores obtained in the experiment varying the translation between the two point clouds in the XY plane. The continuous line inside the box is the median value, and the dashed line is the average. On the left of each box, the points represent the individual score values.

4.4 XY Yaw

The last experiment investigates the effects of both a translation in the XY plane and a yaw rotation. For the translation, the same quadrant strategy is used, but with only $0.5m$ translation for each axis, because all strategies on all quadrants has a reasonable score for this value. For the rotation, each of the four quadrant translations is rotated by 10°, -10°, 30° and -30°, to evaluate the effect of small and big rotations.

Figure 5 show the violin plot for the scores obtained for all combinations in the experiment. It is observed that the base ICP method is unable to converge in various cases, while the covariance weighting strategy is unable to converge in only one case, and the others converge in all cases.

4.5 Final considerations

The experiments explored various cases of initial misalignment between both point clouds in scenarios close to the ones expected in the application, and consistently the weighting strategies showed a smaller instability in the final score of the alignment process. All experiments show that incorporating

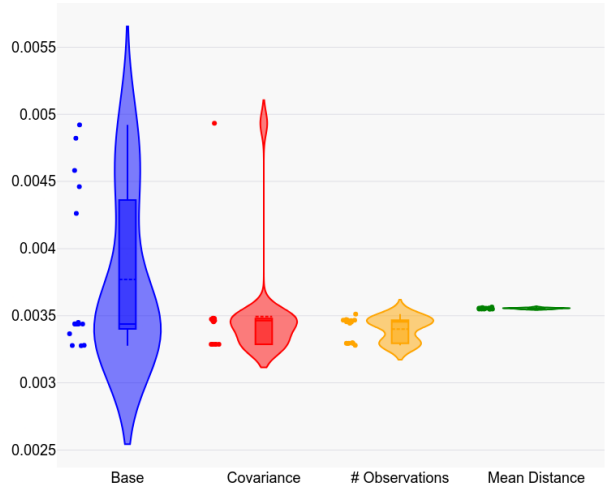


Figure 5. Violin plot of the scores obtained in the experiment varying the translation between the two point clouds in the *XY* plane and the orientation in the Yaw axis. The continuous line inside the box is the median value, and the dashed line is the average. On the left of each box, the points represent the individual score values.

the weighting strategy in the ICP allows a smaller variance of scores to be achieved compared to the base ICP.

Supplementary materials[1] are available with the code used to align the point clouds, the map files used to construct the point clouds and its weights, and detailed tables of all experiments, their scores, number of iterations and duration. Due to space constraints, a detailed analysis of the number of iterations in each experiment was not included in this section, but in the supplementary materials is shown that the usage of the weighting strategies allows the ICP to also need less iterations to converge in various cases, which may reduce the computational cost of the alignment process.

5. Conclusion

This work proposes the usage of three information, that are usually available in feature-based visual SLAM pipelines, to compose a weight that represent the confidence of each map landmark. Then proposes a change in the original ICP data association step to use this weight, attempting to improve the quality of correspondences, and thus improving the convergence of the algorithm.

The experiments show evidence that the usage of the weights might help to improve the convergence of the ICP algorithm with only a simple modification to the data association step. Also, the confidence based on the number of observations showed similar performance, and in some cases even better performance, to the one based on the covariance of the position, while being significantly cheaper to compute, indicating it could potentially be a good replacement for the covariance in this alignment scenario.

[1] https://github.com/CVRLab/weighted-icp-correspondences

Some limitations of this work include the evaluation of the proposed ICP in only one map pair, since different maps produce different scores when aligned and could not be compared directly. Other limitation is the range of misalignment cases tested in the experiments, that are reduced to only what is expected in the application. For these reasons, further statistical analysis is still needed to determine if the weights improve the convergence on all cases and in all kinds of environments.

The main contributions of this work was the deliberation of metrics of the feature-based visual SLAM pipeline that usually are already available in implementations and could be used to represent the confidence of landmarks, the method to incorporate the point confidence as a weighting factor in the ICP without affecting much of the simplicity of the algorithm, and the experimentation with various misalignment cases to have an initial understanding if the weighting strategies might improve the ICP convergence.

Future works might include a more detailed statistical analysis with a wider range of initial alignment transformations and on more environments, to increase the plurality of environmental structures and verify the generalization power of the ICP with the modifications proposed in this work. An extensive comparison between the confidences based on the covariance and the number of observations may also be interesting to determine if their performance for this alignment problem is indeed equivalent. Finally, it may be interesting to explore other common methods and techniques in mobile robotics that have the potential to benefit from incorporating the proposed confidence weights.

Author contributions

Felipe Pires Saraiva: Conceptualization, Methodology, Software, Investigation, Writing - Original Draft, Writing - Review & Editing; Gustavo Teodoro Laureano: Supervision, Conceptualization, Methodology, Writing - Review & Editing; Thiago Henrique de Oliveira: Conceptualization.

References

[1] YOUSIF, K.; BAB-HADIASHAR, A.; HOSEINNEZHAD, R. An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics. *Intelligent Industrial Systems*, v. 1, n. 4, p. 289–311, dez. 2015. ISSN 2363-6912, 2199-854X. Disponível em: ⟨http://link.springer.com/10.1007/s40903-015-0032-7⟩.

[2] SCHLEICHER, D. et al. Real-time hierarchical stereo Visual SLAM in large-scale environments. *Robotics and Autonomous Systems*, v. 58, n. 8, p. 991–1002, ago. 2010. ISSN 09218890. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S0921889010000849⟩.

[3] BALLESTA, M. et al. Multi-robot map alignment in visual SLAM. *WSEAS TRANSACTIONS on SYSTEMS*, v. 9, p. 213–222, fev. 2010.

[4] BESL, P.; MCKAY, N. D. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 14, n. 2, p. 239–256, fev. 1992. ISSN 0162-8828, 2160-9292. Disponível em: ⟨http://ieeexplore.ieee.org/document/121791/⟩.

[5] POMERLEAU, F.; COLAS, F.; SIEGWART, R. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Foundations and Trends® in Robotics*, v. 4, n. 1, p. 1–104, 2015. ISSN 1935-8253, 1935-8261. Disponível em: ⟨http://www.nowpublishers.com/article/Details/ROB-035⟩.

[6] AZZAM, R. et al. Feature-based visual simultaneous localization and mapping: a survey. *SN Applied Sciences*, v. 2, n. 2, p. 224, fev. 2020. ISSN 2523-3963, 2523-3971. Disponível em: ⟨http://link.springer.com/10.1007/s42452-020-2001-3⟩.

[7] GAO, X.; ZHANG, T. *Introduction to Visual SLAM: From Theory to Practice*. Singapore: Springer Singapore, 2021. ISBN 9789811649387 9789811649394. Disponível em: ⟨https://link.springer.com/10.1007/978-981-16-4939-4⟩.

[8] RUSINKIEWICZ, S.; LEVOY, M. Efficient variants of the ICP algorithm. In: *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*. Quebec City, Que., Canada: IEEE Comput. Soc, 2001. p. 145–152. ISBN 978-0-7695-0984-6. Disponível em: ⟨http://ieeexplore.ieee.org/document/924423/⟩.

[9] BOSSE, M. et al. An Atlas framework for scalable mapping. In: *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*. Taipei, Taiwan: IEEE, 2003. p. 1899–1906. ISBN 978-0-7803-7736-3. Disponível em: ⟨http://ieeexplore.ieee.org/document/1241872/⟩.

[10] Zhe Ji et al. Probabilistic 3D ICP algorithm based on ORB feature. In: *2013 IEEE Third International Conference on Information Science and Technology (ICIST)*. Yangzhou, Jiangsu, China: IEEE, 2013. p. 282–282. ISBN 978-1-4673-2764-0 978-1-4673-5137-9 978-1-4673-2763-3. Disponível em: ⟨http://ieeexplore.ieee.org/document/6747555/⟩.

[11] BAI, H. ICP Algorithm: Theory, Practice and Its SLAM-oriented Taxonomy. *Applied and Computational Engineering*, v. 2, n. 1, p. 10–21, mar. 2023. ISSN 2755-2721, 2755-273X. Disponível em: ⟨https://www.ewadirect.com/proceedings/ace/article/view/748⟩.

[12] HUANG, X. et al. *A comprehensive survey on point cloud registration*. arXiv, 2021. Version Number: 2. Disponível em: ⟨https://arxiv.org/abs/2103.02690⟩.

[13] THRUN, S.; BURGARD, W.; FOX, D. *Probabilistic robotics*. Cambridge, Mass: MIT Press, 2005. (Intelligent robotics and autonomous agents). OCLC: ocm58451645. ISBN 978-0-262-20162-9.

[14] HARTLEY, R.; ZISSERMAN, A. *Multiple view geometry in computer vision*. 2nd ed. ed. Cambridge, UK: Cambridge University Press, 2004. OCLC: 171123855. ISBN 978-0-511-18711-7.

[15] RUSU, R. B.; COUSINS, S. 3D is here: Point Cloud Library (PCL). In: *2011 IEEE International Conference on Robotics and Automation*. Shanghai, China: IEEE, 2011. p. 1–4. ISBN 978-1-61284-386-5 978-1-61284-385-8. Disponível em: ⟨https://ieeexplore.ieee.org/document/5980567/⟩.

[16] MUR-ARTAL, R.; TARDOS, J. D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, v. 33, n. 5, p. 1255–1262, out. 2017. ISSN 1552-3098, 1941-0468. Disponível em: ⟨http://ieeexplore.ieee.org/document/7946260/⟩.

[17] STURM, J. et al. A benchmark for the evaluation of RGB-D SLAM systems. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vilamoura-Algarve, Portugal: IEEE, 2012. p. 573–580. ISBN 978-1-4673-1736-8 978-1-4673-1737-5 978-1-4673-1735-1. TUM RGB-D: https://cvg.cit.tum.de/data/datasets/rgbd-dataset. Disponível em: ⟨http://ieeexplore.ieee.org/document/6385773/⟩.