Robust Curvature-Based Feature Descriptors for Noisy Point Cloud Registration

Giorgio De Magistris[1], Mattia Pannone[1]

[1]Department of Computer, Control and Management Engineering, Sapienza University of Rome, Rome, Italy

Abstract
This paper presents an improved methodology for point cloud registration using curvature-based feature descriptors. The core contribution is the adoption of the Umbrella Curvature method for curvature estimation, evaluated against the standard Surface Variance approach under various noise conditions. To enhance robustness, local variance features are introduced alongside curvature, forming a composite descriptor for more reliable point matching. Registration is performed using a modified Iterative Closest Point (ICP) algorithm leveraging these features. Extensive experiments demonstrate that Umbrella Curvature significantly improves alignment accuracy, particularly under high noise, and that the proposed feature aggregation further enhances robustness. Some limitations in specific geometric configurations are discussed.

1. Introduction

Point cloud processing has become increasingly critical in various fields, including computer vision, robotics, and 3D modeling. Point clouds are typically generated by 3D scanners, representing objects or scenes as discrete sets of spatial points. In many applications, multiple point clouds must be aligned or merged into a coherent global model, a process known as registration.

In the 2D image domain, local feature descriptors such as SIFT [1], HOG [2], and LBP [3] are widely used for matching and alignment tasks [4, 5, 6] and in the field of robotics [7, 8]. Analogously, in 3D space, feature descriptors that capture local geometric properties are crucial for establishing correspondences between points across different scans.

Among various geometric features, curvature plays a particularly important role due to its invariance under rigid transformations (translations and rotations). Curvature characterizes the local shape of a surface, offering discriminative information that is stable across different poses of the same object.

In this work, we explore curvature-based registration of point clouds using the Umbrella Curvature [9] method, a recently proposed technique that computes curvature based on the homogeneous distribution of neighboring points, mimicking the structure of an umbrella. This method is compared against the classical Surface Variance approach [10, 11], which estimates curvature from the eigenvalues of the local covariance matrix.

Beyond curvature alone, we introduce local variance—the statistical dispersion of neighboring points—as an additional feature to improve matching reliability. Combining curvature and variance provides a richer descriptor that helps distinguish between points with similar curvature values but differing local geometries.

A major focus of this study is the robustness of curvature estimation in the presence of noise. Noise is common in real-world scans due to sensor imperfections and environmental factors, and it can significantly degrade feature reliability. To address this, we propose two median-based strategies to enhance the stability of curvature estimation without resorting to traditional denoising techniques.

Finally, we integrate these feature descriptors into a customized version of the Iterative Closest Point (ICP) algorithm [12, 13, 14], aligning point clouds based on feature similarity rather than pure spatial proximity. Extensive experiments are conducted on publicly available datasets, evaluating accuracy under various conditions including high noise levels.

The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 details the methodology; Section 4 presents experimental results and analysis; Section 5 concludes with key findings and future directions.

2. Related Works

Feature-based registration of point clouds is a well-established research area with significant attention in recent years. Several approaches have been proposed for extracting geometric descriptors, estimating curvature, selecting neighboring points, and performing robust registration.

Curvature Estimation. Foorginejad and Aghbari [9] introduced the Umbrella Curvature method, which employs homogeneously distributed neighbors to improve the robustness of curvature estimation. The method calculates the local normal and curvature of a point using an umbrella-like arrangement of neighbors, providing better

ICYRIME 2025: 10th International Conference of Yearly Reports on Informatics, Mathematics, and Engineering. Czestochowa, January 14-16, 2025
demagistris@diag.uniroma1.it (G. D. Magistris)

performance compared to traditional eigenvalue-based techniques.

Earlier, Pauly et al. [10] proposed the widely-used Surface Variation descriptor, estimating curvature as the ratio between the smallest eigenvalue and the sum of all eigenvalues of the covariance matrix built from local neighborhoods. This approach is simpler and computationally efficient, but can be less robust in regions with uneven point distributions.

Alternative curvature estimation strategies include fitting local surfaces to point sets. Zhang et al. [15] proposed estimating curvature by fitting normal sections in multiple directions, offering improved stability compared to direct covariance analysis. Such methods, however, typically incur higher computational costs.

Neighbor Selection. Accurate curvature estimation strongly depends on the choice of neighboring points. Park et al. [16] introduced the Elliptic Gabriel Graph (EGG) for selecting neighbors that better capture local geometric relationships. Friedman et al. [17] proposed the KD-Tree data structure, which remains a standard for efficiently querying nearest neighbors, though it does not guarantee uniform angular distribution. To overcome this, homogeneous neighborhood selection—favoring angularly well-distributed points—has been proposed, especially for methods like Umbrella Curvature.

Point Cloud Feature Extraction. Beyond hand-crafted features, learning-based methods such as Xiang et al. [18] used multilayer perceptrons to directly learn point descriptors from local neighborhoods reordered via space-filling curves. Although effective, such approaches are typically more complex and suited to large-scale datasets.

Registration Algorithms. The Iterative Closest Point (ICP) algorithm [12] remains the foundational method for rigid point cloud registration, minimizing point-wise distances iteratively. Variants such as He et al. [19] have incorporated geometric features, including curvature, normals, and density, to improve matching accuracy. Yao et al. [20] introduced a similarity measure based on local curvature variations, demonstrating enhanced robustness compared to purely geometric methods.

Other works, like Bae and Lichti [21], proposed efficient feature-based registration pipelines that account for noise and missing data. Belton and Lichti [22] explored local variance as a metric for classifying and segmenting point clouds, an idea that inspires the use of variance as an auxiliary descriptor in the current work.

Summary. Overall, while many approaches leverage curvature and variance for analysis, the combination of umbrella curvature estimation, local variance filtering, and noise-robust feature aggregation within an ICP framework remains relatively unexplored. This paper aims to systematically study and validate these contributions under noisy conditions.

3. Implementation

This section outlines the full pipeline developed for feature extraction, curvature estimation, and point cloud registration. The process includes data acquisition, neighborhood selection, curvature and variance computation, noise robustness strategies, and an Iterative Closest Point (ICP) algorithm enhanced with geometric features.

All components were implemented in Python and tested on Google Colab, with the code structured into modular functions to ensure clarity and reproducibility. Public datasets and annotated notebooks were used to facilitate consistent experimentation.

3.1. Data Acquisition

The datasets were sourced from the Stanford 3D Scanning Repository[1], a well-known benchmark containing high-resolution 3D scans. Three objects were selected—Bunny, Armadillo, and Dragon—and for each, four different scans were used. One scan served as the target, while the remaining three were used as sources.

The point clouds, stored in `.ply` format, consist of (x, y, z) coordinates. Unless otherwise stated (e.g., in downsampling experiments), raw data was used to preserve geometric detail. Figure **??** shows visualizations of the selected objects.

3.2. Neighbor Selection

Curvature and variance estimation rely on meaningful neighborhood selection. We employed two strategies: a KD-Tree-based search for retrieving the k nearest neighbors by Euclidean distance, and a refined selection based on spatial distribution for the Umbrella method.

In the latter, neighbors were filtered by cosine similarity to ensure even angular distribution around the query point. A threshold of 0.8 was used to retain only those neighbors well-distributed relative to the mean direction vector. Figure **??** illustrates the difference between standard and homogeneous neighborhoods.

PCA was then applied to each neighborhood: points were centered, the covariance matrix was computed, and eigenvalues and eigenvectors extracted. The eigenvector corresponding to the smallest eigenvalue was taken as the estimated surface normal, while the eigenvalues were used in curvature and variance computation.

3.3. Curvature Estimation

We implemented and compared two curvature estimation methods: Surface Variance Curvature and Umbrella Curvature.

[1]http://graphics.stanford.edu/data/3Dscanrep/

Surface Variance Curvature [10] is computed from PCA eigenvalues:

$$k_{\text{SV}} = \frac{\lambda_{\text{min}}}{\sum_{i=1}^{3} \lambda_i} \tag{1}$$

Umbrella Curvature [9] instead computes directional deviation along the surface normal:

$$k_{\text{UM}} = \sum_{i=1}^{N} \left| \frac{N_i - p}{\|N_i - p\|} \cdot n \right| \tag{2}$$

Here, p is the query point, N_i are neighboring points, and n is the surface normal. The normalized displacement vectors are projected onto the normal vector, and their absolute projections are summed to obtain the curvature value.

Figure **??** shows that Umbrella Curvature provides sharper contrast in areas with high geometric detail, making it more suitable for feature-based registration.

3.4. Robustness to Noise

To assess robustness, Gaussian noise was synthetically added to point clouds. Two noise-resilient curvature methods were developed.

The first, Umbrella-2, replaces the mean center with the median of the neighborhood:

$$\text{new\_points} = \text{neighbors} - \text{median}(\text{neighbors})$$

The second, Umbrella-3, applies a sliding window median filter to the neighborhood:

$points \leftarrow$ neighbors of p
Initialize $window$
Initialize $points_{\text{filtered}}$
for i in range of neighbors **do**
    Update $window$ with $points[i]$
    $points_{\text{filtered}}[i] \leftarrow \text{median}(window)$
end for

These methods leverage the statistical robustness of the median to suppress noise while preserving geometric features.

3.5. Registration Algorithm

Point cloud alignment was performed using a modified version of ICP. Traditional ICP uses proximity-based correspondence; we instead introduced feature-based matching using curvature and local variance.

In the first variant (ICP-Curvature), for each source point p_s, k nearest target neighbors are found. The neighbor p_t with the most similar curvature is selected if the difference is below a threshold τ_1.

In the second variant (ICP-Curvature + Local Variance), after curvature-based selection, local variance is also compared. The match is accepted only if the variance difference is below a second threshold τ_2.

The two algorithms are summarized below:

Rigid transformations were computed using Singular Value Decomposition (SVD). After computing and centering the centroids of the correspondence pairs, the covariance matrix was formed, and SVD was applied:

$$T = \begin{bmatrix} R & t \\ \mathbf{0}^T & 1 \end{bmatrix}$$

Convergence was reached when the registration error, defined as the mean Euclidean distance between matched pairs, stabilized or when a maximum number of iterations was exceeded.

3.6. Parameter Selection

To balance accuracy, robustness, and speed, parameters were chosen based on prior work and empirical tuning. For curvature estimation, $k = 8$ neighbors were used. In the Umbrella method, cosine similarity ensured even distribution. ICP correspondence search used $k = 50$ neighbors, while local variance was computed using 1000 points.

Thresholds for feature matching were set to $\tau_1 = \tau_2 = 0.0001$, based on analysis of feature value distributions. The median filter window for Umbrella-3 was set to size 5. ICP typically ran for 50–100 iterations, with early stopping if no improvement was observed. Some tests downsampled 50% of the points to study efficiency.

In summary, the implementation emphasized modularity, geometric fidelity, and resilience to noisy conditions while remaining efficient and reproducible.

4. Experimental Results

To validate the effectiveness of the proposed methods, we conducted a series of experiments on several real-world point clouds. These experiments used both clean and synthetically noised data, enabling evaluation under diverse conditions. We focused on quantitative comparisons and visual inspections to assess registration quality.

We used the Stanford 3D Scanning Repository [23], selecting four scans per object (Bunny, Armadillo, Dragon). One scan was chosen as the target, while the other three served as sources. Registration performance was measured using the mean Euclidean distance between matched point pairs (registration error) and visual overlap of aligned scans.

A key comparison was made between two curvature estimation strategies: Umbrella Curvature and Surface

Algorithm 1 ICP using Curvature Feature

Initialize empty correspondence sets.
for each point p_s in source point cloud **do**
    Find k nearest neighbors of p_s in target point cloud.
    For each neighbor, compute curvature difference.
    Select neighbor p_t with minimum curvature difference.
    if curvature difference $\leq \tau_1$ **then**
        Add (p_s, p_t) to correspondence set.
    end if
end for
Estimate optimal transformation from correspondences.
Apply transformation to source cloud.
Repeat until convergence.

Algorithm 2 ICP using Curvature and Local Variance

Initialize empty correspondence sets.
for each point p_s in source point cloud **do**
    Find k nearest neighbors of p_s in target point cloud.
    For each neighbor, compute curvature difference.
    Select neighbor p_t with minimum curvature difference.
    if curvature difference $\leq \tau_1$ **then**
        Compute local variance at p_s and p_t.
        if variance difference $\leq \tau_2$ **then**
            Add (p_s, p_t) to correspondence set.
        end if
    end if
end for
Estimate optimal transformation from correspondences.
Apply transformation to source cloud.
Repeat until convergence.

Variation. As summarized in Tables 1 to 3, Umbrella Curvature generally produced lower registration errors and required fewer ICP iterations. Visualizations in Figures 1 to 9 confirm that this method better preserves geometric details during alignment. Nonetheless, failure cases occurred when the initial misalignment was large or when different regions of the object had similar curvature characteristics—such as the Bunny's head and back.

To evaluate computational efficiency, we downsampled the point clouds to 50% of their original size. As shown in Table 4, this introduced only a small increase in error, suggesting that downsampling offers a practical trade-off between speed and accuracy.

We also explored enhancements to the base curvature descriptor. Adding a local variance feature improved registration accuracy in most cases, particularly in regions with ambiguous curvature but differing point dispersion (see Tables 5 to 7). For the Dragon model, however, the effect was inconsistent, indicating that the benefit may be data-dependent.

Another important factor was the choice of neighborhood size in the curvature computation. Experiments using $k = 8$ and $k = 100$ neighbors (Tables 8 to 10) showed that while a larger neighborhood occasionally led to faster convergence, it also risked oversmoothing the geometry. We therefore adopted $k = 8$ as the default to maintain local detail.

Finally, we tested robustness to noise by adding Gaussian perturbations to the source point clouds. Variants of the Umbrella curvature method—including median centering (Umbrella2), local filtering (Umbrella3), and the combination of Umbrella3 with local variance—were compared. Results, shown in Tables 11 to 13 and Figures 10 to 12, indicate that both Umbrella2 and Umbrella3 improve noise resilience. The most robust variant, Umbrella3 with local variance, achieved very low registration errors in several cases, although some visually incorrect alignments occurred under heavy noise—highlighting the limits of local variance in such conditions.

In summary, the experiments demonstrate that Umbrella curvature outperforms Surface Variation for point cloud registration, particularly when combined with local variance or median filtering. These improvements remain effective across clean, downsampled, and noisy data, confirming the proposed method's robustness and accuracy.

5. Conclusions

This work investigated the use of curvature-based features to enhance point cloud registration, particularly under challenging conditions such as noise and partial data. The proposed approach combined Umbrella Curvature estimation with local variance filtering, and extended the traditional ICP algorithm using feature-driven matching strategies.

The experimental results highlighted several key findings. Umbrella Curvature consistently outperformed Surface Variation in providing reliable and discriminative geometric descriptors, leading to lower registration errors and faster convergence. The addition of local variance proved useful in distinguishing between regions with similar curvature, especially when those regions originated from different parts of the object. Moreover, the robustness of the method was significantly improved by incorporating median-based centering and local filtering prior to curvature estimation. Even when point clouds were downsampled to half their original size, the registration remained accurate and efficient, demonstrating the method's scalability.

Despite these promising results, some limitations were observed. A low numerical registration error did not always imply correct alignment, particularly in cases involving large initial misalignments or symmetric geometries. Additionally, while local variance was helpful in clean data scenarios, its effectiveness diminished under strong noise. The method also encountered difficulties when dealing with large flat surfaces or repeated structures that exhibit similar curvature patterns.

Looking ahead, future work will explore the integration of global geometric descriptors or semantic segmentation to improve performance in ambiguous regions. Adaptive strategies for selecting neighborhood sizes based on local point density and noise characteristics may also enhance accuracy. Furthermore, extending the method to non-rigid registration tasks—where local deformations occur—presents an exciting direction. Finally, learning-based curvature estimation could be investigated as a means to further improve robustness and generalization.

In conclusion, the results confirm that combining well-designed curvature descriptors with local statistical features can lead to substantial improvements in point cloud registration, even under noisy and incomplete conditions.

6. Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] D. G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the seventh IEEE international conference on computer vision, volume 2, Ieee, 1999, pp. 1150–1157.

[2] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), volume 1, Ieee, 2005, pp. 886–893.

[3] T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, S. Huovinen, Outex-new framework for empirical evaluation of texture analysis algorithms, in: 2002 international conference on pattern recognition, volume 1, IEEE, 2002, pp. 701–706.

[4] F. Bonanno, G. Capizzi, S. Coco, C. Napoli, A. Laudani, G. L. Sciuto, Optimal thicknesses determination in a multilayer structure to improve the spp efficiency for photovoltaic devices by an hybrid fem - cascade neural network based approach, in: 2014 International Symposium on Power Electronics, Electrical Drives, Automation and Motion, SPEEDAM 2014, 2014, p. 355 – 362. doi:10.1109/SPEEDAM.2014.6872103.

[5] N. Brandizzi, A. Fanti, R. Gallotta, S. Russo, L. Iocchi, D. Nardi, C. Napoli, Unsupervised pose estimation by means of an innovative vision transformer, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 13589 LNAI, 2023, p. 3 – 20. doi:10.1007/978-3-031-23480-4_1.

[6] E. Iacobelli, S. Russo, C. Napoli, A machine learning based real-time application for engagement detection, in: CEUR Workshop Proceedings, volume 3695, 2023, p. 75 – 84.

[7] C. Napoli, C. Napoli, V. Ponzi, A. Puglisi, S. Russo, I. E. Tibermacine, Exploiting robots as healthcare resources for epidemics management and support caregivers, in: CEUR Workshop Proceedings, volume 3686, 2024, p. 1 – 10.

[8] C. Randieri, A. Pollina, A. Puglisi, C. Napoli, Smart glove: A cost-effective and intuitive interface for

		Iterations	Registration Error
Test 1	Umbrella Method	19	0.0025711
	Surface Variance	47	0.0027847
Test 2	Umbrella Method	13	0.0082389
	Surface Variance	15	0.0093368
Test 3	Umbrella Method	21	0.0079683
	Surface Variance	47	0.0093608

Table 1
This table shows the execution of the ICP algorithm on 3 different models of the Bunny object, you can see the comparison between the use of the Surface Variance method and that of the Umbrella Curvature.

		Iterations	Registration Error
Test 1	Umbrella Method	35	0.0039208
	Surface Variance	40	0.0037737
Test 2	Umbrella Method	17	0.009969
	Surface Variance	56	0.0122429
Test 3	Umbrella Method	21	0.0093331
	Surface Variance	29	0.0134528

Table 2
This table shows the execution of the ICP algorithm on 3 different models of the Armadillo object, you can see the comparison between the use of the Surface Variance method and that of the Umbrella Curvature.

		Iterations	Registration Error
Test 1	Umbrella Method	38	0.0022854
	Surface Variance	43	0.0022744
Test 2	Umbrella Method	33	0.0035608
	Surface Variance	36	0.0042893
Test 3	Umbrella Method	18	0.0068711
	Surface Variance	43	0.0061069

Table 3
This table shows the execution of the ICP algorithm on 3 different models of the Dragon object, you can see the comparison between the use of the Surface Variance method and that of the Umbrella Curvature.

		Iterations	Registration Error
Bunny	Umbrella Method	19	0.003647
	Surface Variance	39	0.0036281
Armadillo	Umbrella Method	13	0.0096267
	Surface Variance	41	0.0049265
Dragon	Umbrella Method	48	0.0029148
	Surface Variance	36	0.0031384

Table 4
This table shows the execution of the ICP algorithm on all 3 objects (with only 2 point clouds for each object) using reduced size point clouds, i.e. randomly deleting some points (in order to reduce the execution time).

		Iterations	Registration Error
Test 1	Umbrella Method	19	0.0025711
	Umbrella + Local Variance	32	0.0018554
Test 2	Umbrella Method	13	0.0082389
	Umbrella + Local Variance	14	0.0044293
Test 3	Umbrella Method	21	0.0079683
	Umbrella + Local Variance	18	0.0043945

Table 5
This table shows the execution of the ICP algorithm on 3 different tests of the Bunny object and compares the results achieved with only the use of the curve as a feature Algorithm [**??**]] and that with the addition of the local variance Algorithm [**??**].

		Iterations	Registration Error
Test 1	Umbrella Method	35	0.0039208
	Umbrella + Local Variance	10	0.0024387
Test 2	Umbrella Method	17	0.009969
	Umbrella + Local Variance	23	0.0032814
Test 3	Umbrella Method	21	0.0093331
	Umbrella + Local Variance	14	0.0024006

Table 6
This table shows the execution of the ICP algorithm on 3 different tests of the Armadillo object and compares the results achieved with only the use of the curve as a feature Algorithm [**??**] and that with the addition of the local variance Algorithm [**??**]].

		Iterations	Registration Error
Test 1	Umbrella Method	38	0.0022854
	Umbrella + Local Variance	12	0.0059879
Test 2	Umbrella Method	33	0.0035608
	Umbrella + Local Variance	16	0.0060449
Test 3	Umbrella Method	18	0.0068711
	Umbrella + Local Variance	20	0.0061974

Table 7
This table shows the execution of the ICP algorithm on 3 different tests of the Dragon object and compares the results achieved with only the use of the curve as a feature Algorithm [**??**] and that with the addition of the local variance Algorithm [**??**]].

		Iterations	Registration Error
Test 1	Umbrella with 8 neighbors	19	0.0025711
	Umbrella with 100 neighbors	18	0.0025364
Test 2	Umbrella with 8 neighbors	13	0.0082389
	Umbrella with 100 neighbors	15	0.0090874
Test 3	Umbrella with 8 neighbors	21	0.0079683
	Umbrella with 100 neighbors	10	0.0051346

Table 8
This table shows the execution of the ICP algorithm on 3 different tests of the Bunny object and compares the results achieved with the calculation of the Umbrella curvature with 8 neighbors and the calculation of the same with 100 neighbors.

		Iterations	Registration Error
Test 1	Umbrella with 8 neighbors	35	0.0039208
	Umbrella with 100 neighbors	15	0.0076229
Test 2	Umbrella with 8 neighbors	17	0.009969
	Umbrella with 100 neighbors	11	0.0056058
Test 3	Umbrella with 8 neighbors	21	0.0093331
	Umbrella with 100 neighbors	8	0.0130134

Table 9
This table shows the execution of the ICP algorithm on 3 different tests of the Armadillo object and compares the results achieved with the calculation of the Umbrella curvature with 8 neighbors and the calculation of the same with 100 neighbors.

		Iterations	Registration Error
Test 1	Umbrella with 8 neighbors	38	0.0022854
	Umbrella with 100 neighbors	20	0.0033579
Test 2	Umbrella with 8 neighbors	33	0.0035608
	Umbrella with 100 neighbors	12	0.0048088
Test 3	Umbrella with 8 neighbors	18	0.0068711
	Umbrella with 100 neighbors	12	0.0047976

Table 10
This table shows the execution of the ICP algorithm on 3 different tests of the Dragon object and compares the results achieved with the calculation of the Umbrella curvature with 8 neighbors and the calculation of the same with 100 neighbors.

	Iteration	Registration error
Noise + Umbrella	15	0.0072602
Noise + Umbrella2	14	0.0048573
Noise + Umbrella3	13	0.0050148
Noise + Umbrella3 + local variance	9	0.0012773

Table 11
This table shows the execution of the ICP algorithm between two point clouds of the Bunny object where Guassian noise has been added to one of them, the source point cloud. The results show the execution after calculating: in the first row the curvature with the classic umbrella method; in the second by subtracting the median from the points (umbrella2); in the third row by applying a small filter to the points (umbrella3); in the last one as in the previous one but adding the control on the local variance (algorithm 2).

	Iteration	Registration error
Noise + Umbrella	25	0.0086797
Noise + Umbrella2	9	0.0098761
Noise + Umbrella3	21	0.0079397
Noise + Umbrella3 + local variance	2	0.0021998

Table 12
This table shows the execution of the ICP algorithm between two point clouds of the Armadillo object where Guassian noise has been added to one of them, the source point cloud. The results show the execution after calculating: in the first row the curvature with the classic umbrella method; in the second by subtracting the median from the points (umbrella2); in the third row by applying a small filter to the points (umbrella3); in the last one as in the previous one but adding the control on the local variance (algorithm 2).

	Iteration	Registration error
Noise + Umbrella	11	0.0100419
Noise + Umbrella2	27	0.0052735
Noise + Umbrella3	24	0.004230
Noise + Umbrella3 + local variance	-	-

Table 13
This table shows the execution of the ICP algorithm between two point clouds of the Dragon object where Guassian noise has been added to one of them, the source point cloud. The results show the execution after calculating: in the first row the curvature with the classic umbrella method; in the second by subtracting the median from the points (umbrella2); in the third row by applying a small filter to the points (umbrella3); in the last one as in the previous one but adding the control on the local variance (Algorithm 2).

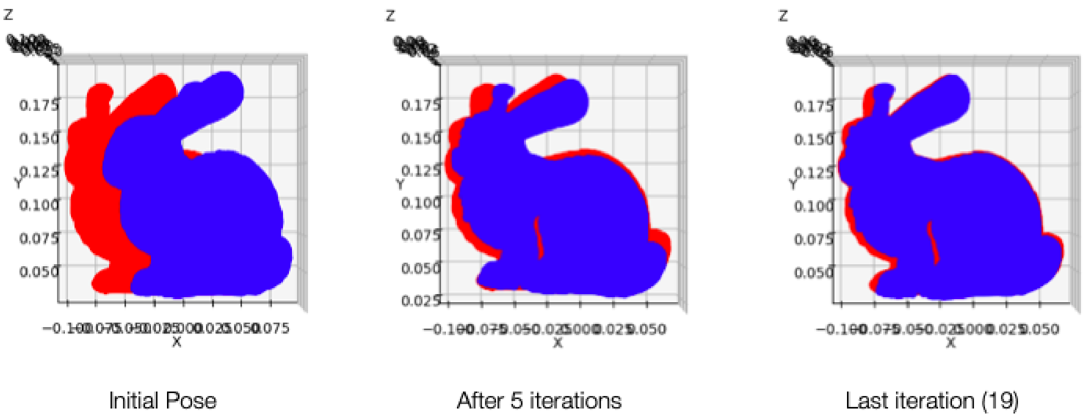

Figure 1: Test 1 of Bunny registration.

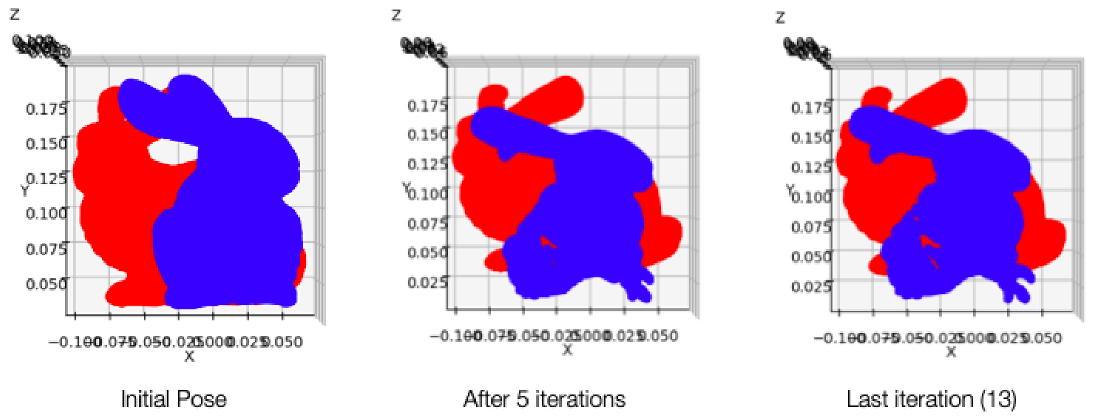

Figure 2: Test 2 of Bunny registration.

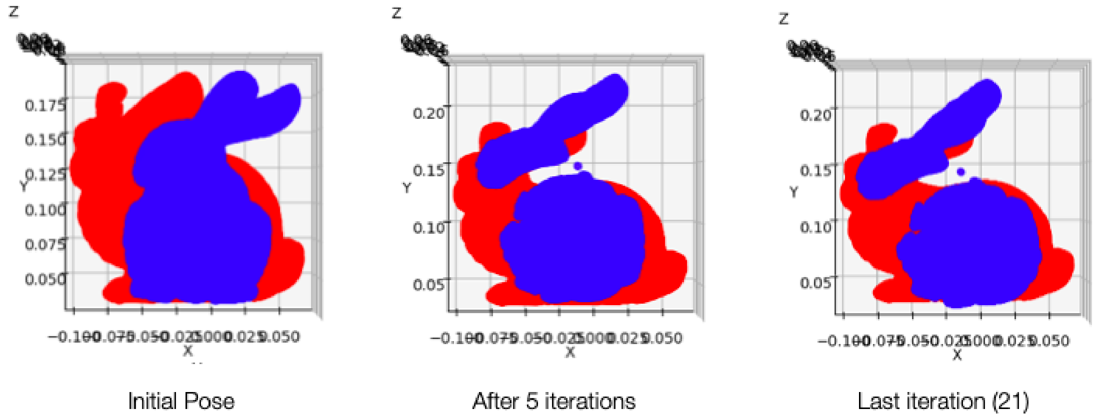

Figure 3: Test 3 of Bunny registration.

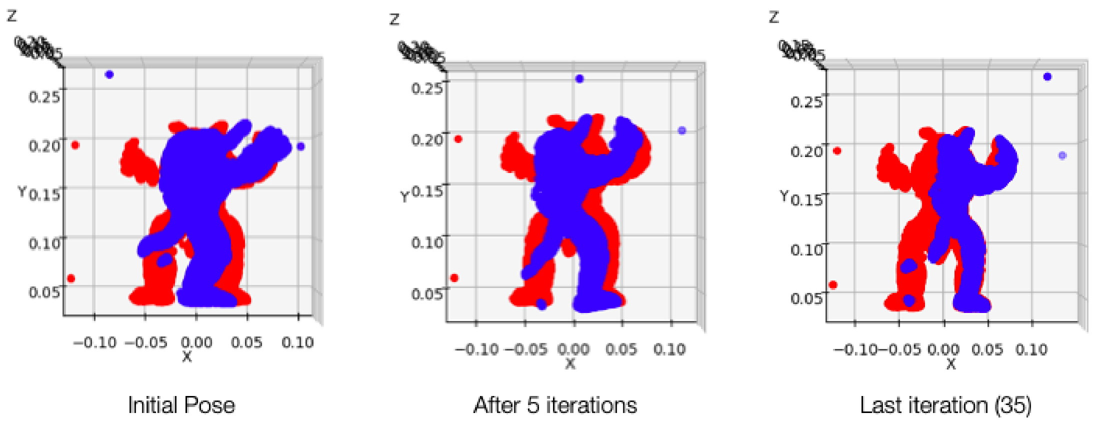

Figure 4: Test 1 of Armadillo registration.

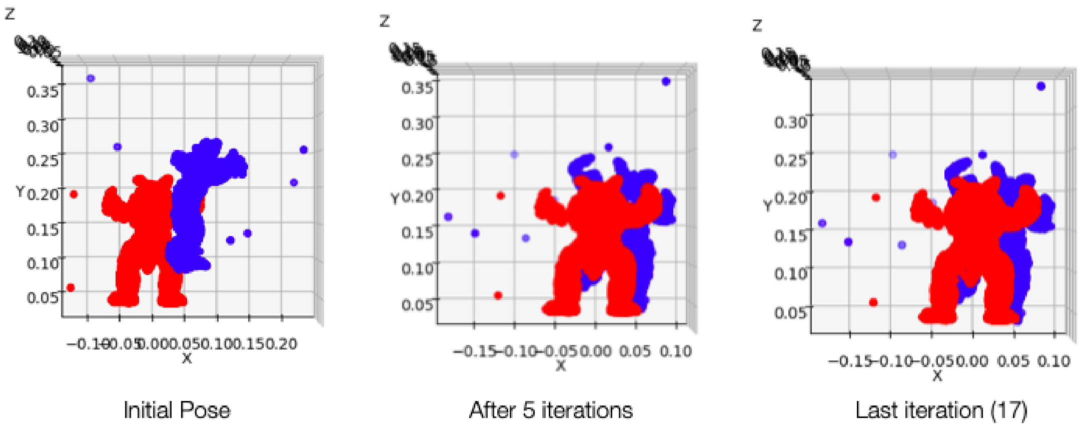

Figure 5: Test 2 of Armadillo registration.

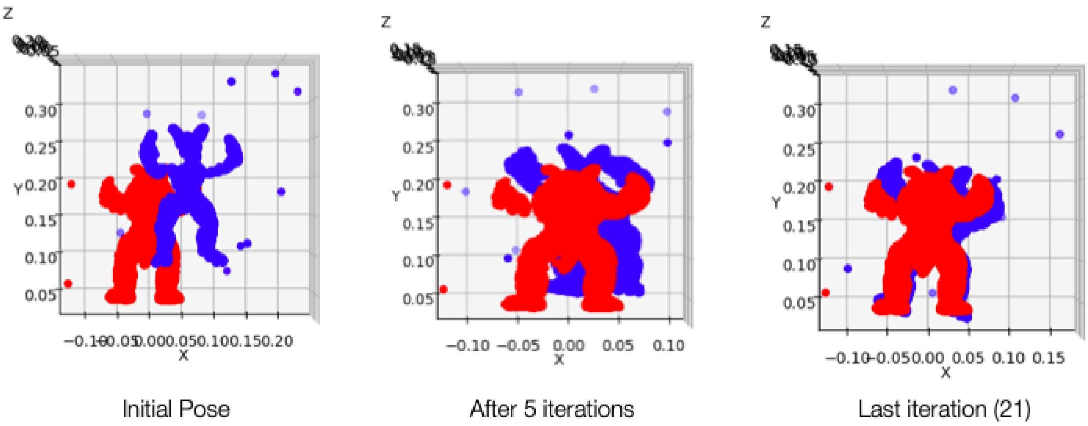

Figure 6: Test 3 of Armadillo registration.

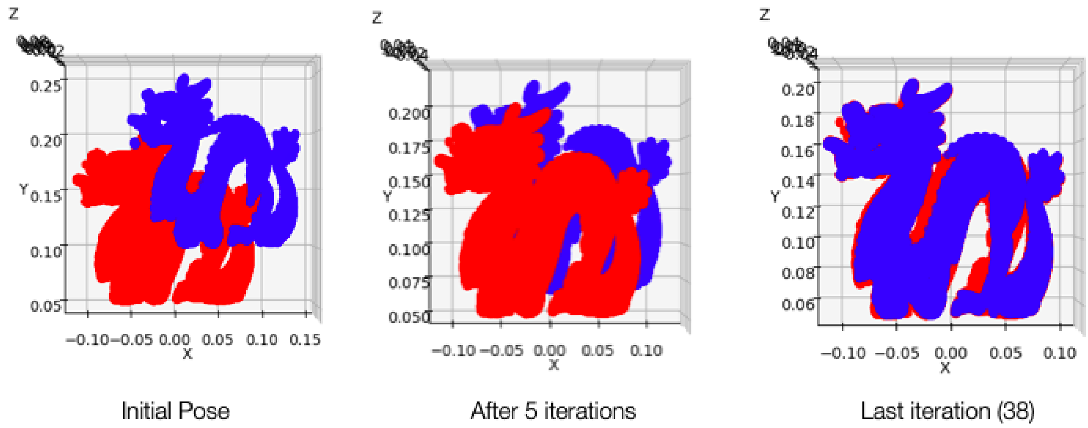

Figure 7: Test 1 of Dragon registration.

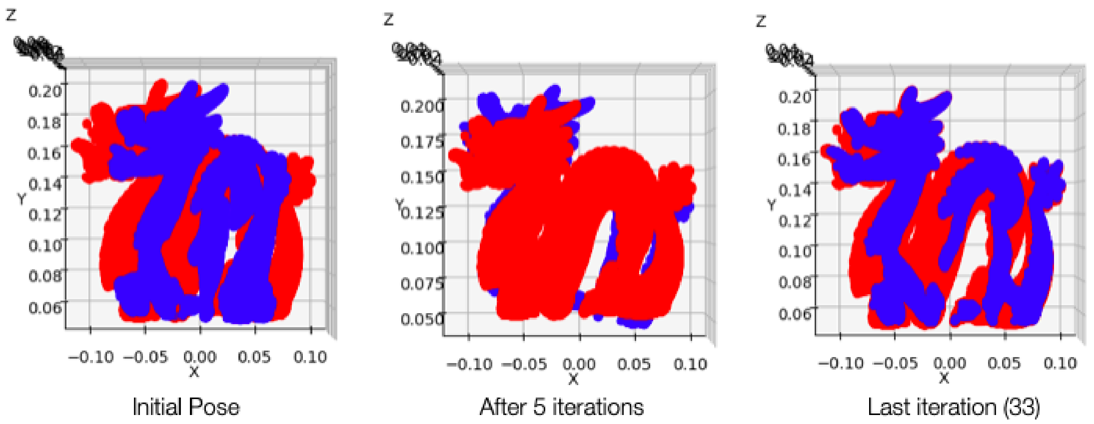

Figure 8: Test 2 of Dragon registration.

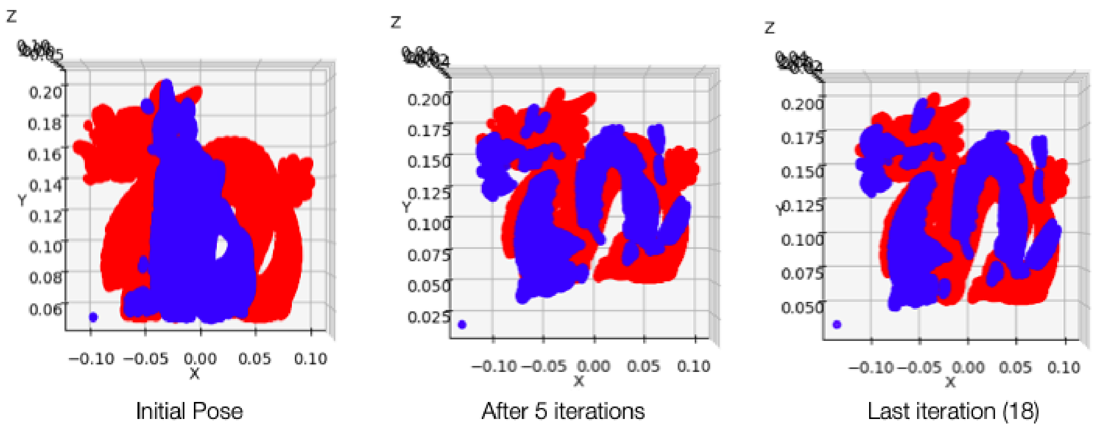

Figure 9: Test 3 of Dragon registration.

advanced drone control, Drones 9 (2025). doi:10.3390/drones9020109.

[9] A. Foorginejad, K. Khalili, Umbrella curvature: A new curvature estimation method for point clouds, Procedia Technology 12 (2014) 347–352.

[10] M. Pauly, M. Gross, L. P. Kobbelt, Efficient simplification of point-sampled surfaces, ACM Transactions on Graphics (TOG) 22 (2003) pp. 669–677.

[11] G. Lo Sciuto, G. Capizzi, R. Shikler, C. Napoli, Organic solar cells defects classification by using a new feature extraction algorithm and an ebnn with an innovative pruning algorithm, International Journal of Intelligent Systems 36 (2021) 2443 – 2464. doi:10.1002/int.22386.

[12] P. J. Besl, N. D. McKay, A method for registration of 3-d shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (1992).

[13] G. Capizzi, G. L. Sciuto, C. Napoli, E. Tramontana, A multithread nested neural network architecture to model surface plasmon polaritons propagation, Micromachines 7 (2016). doi:10.3390/mi7070110.

[14] G. Capizzi, G. L. Sciuto, C. Napoli, R. Shikler, M. Wozniak, Optimizing the organic solar cell manufacturing process by means of afm measurements and neural networks, Energies 11 (2018). doi:10.3390/en11051221.

[15] X. Zhang, H. Li, Z. Cheng, Curvature estimation of 3d point cloud surfaces through the fitting of normal section curvatures, Pattern Recognition 45 (2012) pp. 3250–3263.

[16] J. C. Park, H. Shin, B. K. Choi, Elliptic gabriel graph for finding neighbors in a point set and its application to normal vector estimation, Computer-Aided Design 38 (2006) pp. 975–986.

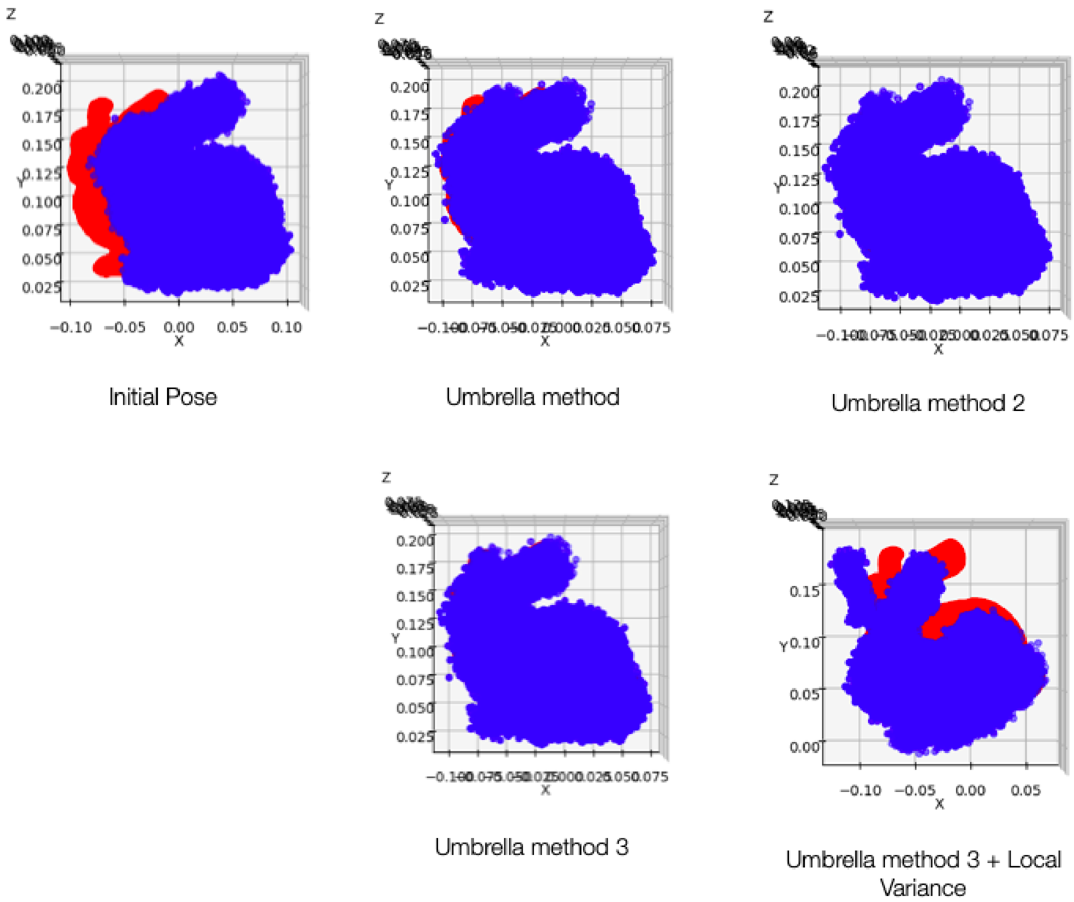

Figure 10: Test of Bunny registration on the point cloud with noise and the different methods to deal with it.

[17] J. H. Friedman, J. L. Bentley, R. Finkel, An algorithm for finding best matches in logarithmic expected time, ACM Transactions on Mathematical Software (TOMS) 3 (1977) pp. 209–226.

[18] X. Xiang, L. Wang, W. Zong, G. Li, Extraction of local structure information of point clouds through space-filling curve for semantic segmentation, ISPRS Journal of Photogrammetry and Remote Sensing 122 (2016) pp. 108–123.

[19] Y. He, B. Liang, J. Yang, S. Li, J. He, An iterative closest points algorithm for registration of 3d laser scanner point clouds with geometric features, Sensors 17 (2017). doi:10.3390/s17081862.

[20] Z. Yao, Q. Zhao, X. Li, Q. Bi, Point cloud registration algorithm based on curvature feature similarity, Measurement 177 (2021) 109274.

[21] K.-H. Bae, D. D. Lichti, A method for automated registration of unorganised point clouds, ISPRS Journal of Photogrammetry & Remote Sensing 63 (2008) 36–54.

[22] D. Belton, D. D. Lichti, Classification and segmentation of terrestrial laser scanner point clouds using local variance information, in: ISPRS Commission V Symposium 'Image Engineering and Vision Metrology', 2006.

[23] B. Curless, M. Levoy, Stanford 3d scanning repository, http://graphics.stanford.edu/data/3Dscanrep/, 1996. Accessed: 2025-05-22.

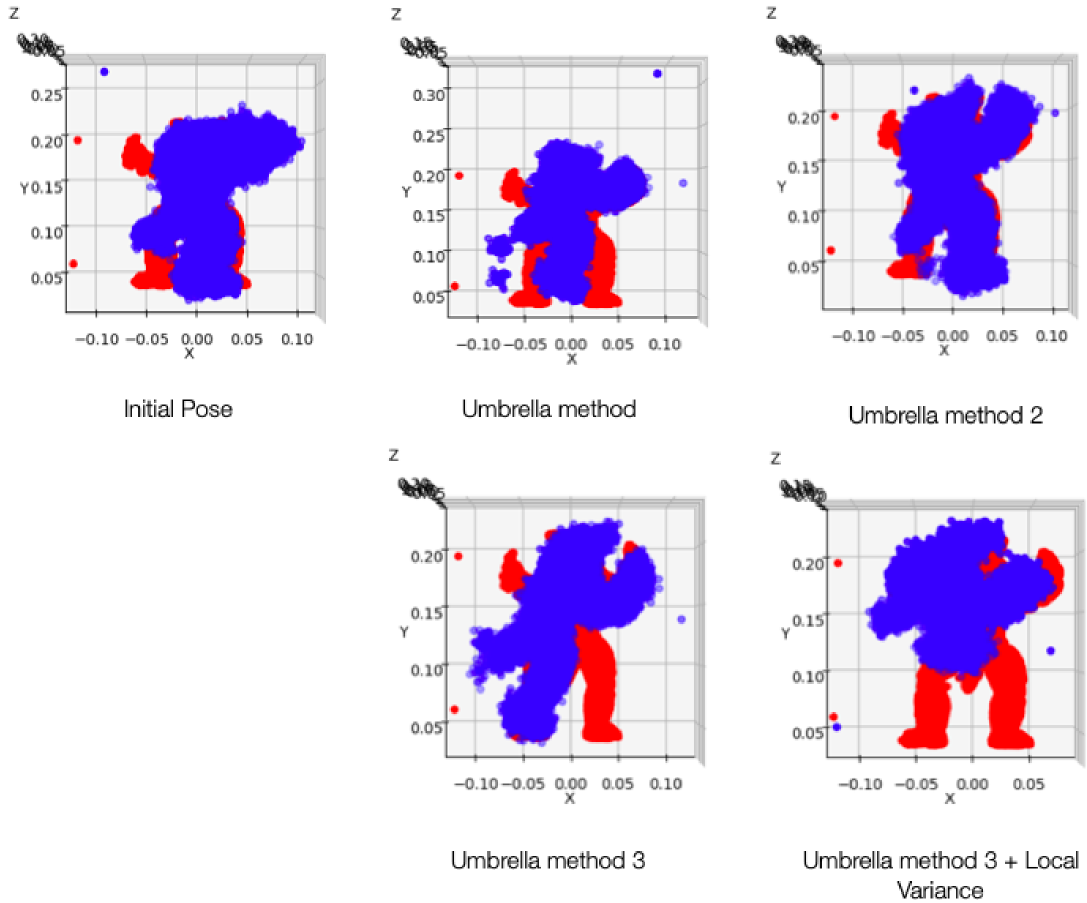

Figure 11: Test of Armadillo registration on the point cloud with noise and the different methods to deal with it.

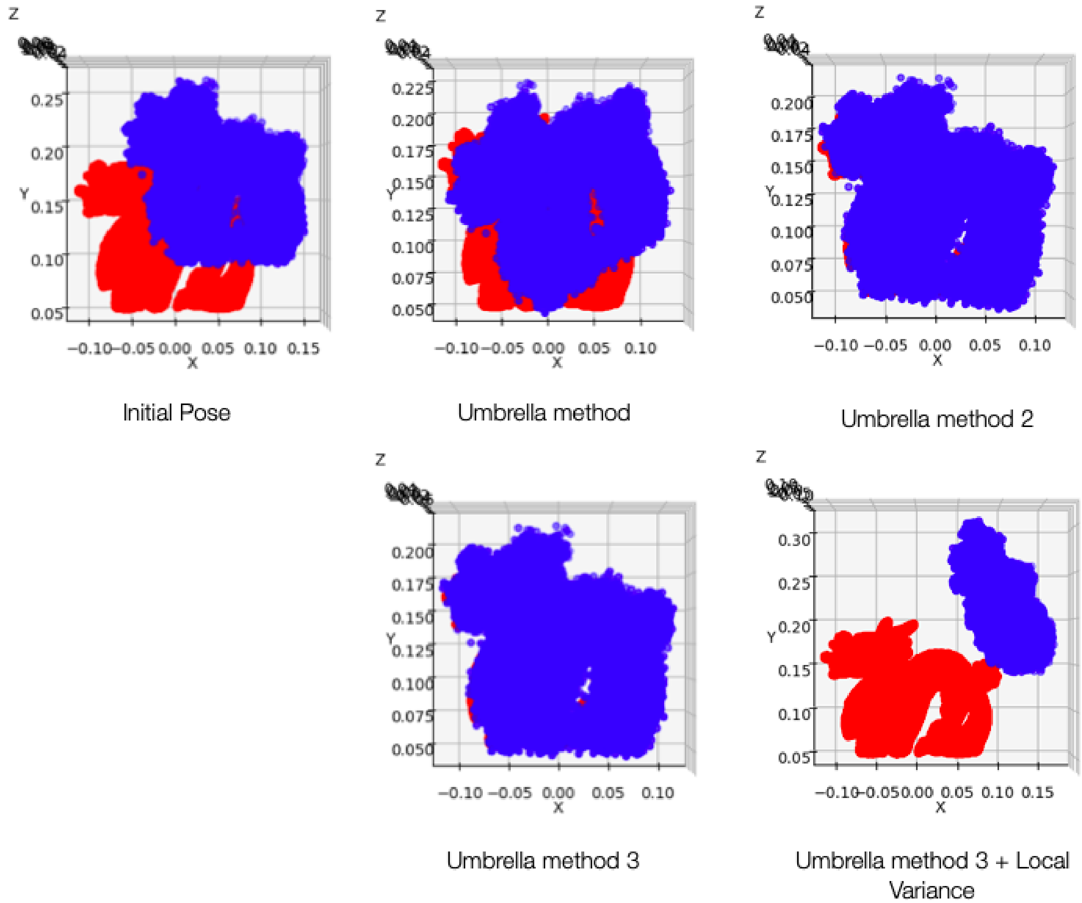

Figure 12: Test of Dragon registration on the point cloud with noise and the different methods to deal with it.