International Journal of Image and Graphics

A Shape-based Weighting Strategy Applied to the Covariance Estimation on ICP

F. A. A. Yamada[1], L. W. X. Cejnog[2], M. B. Vieira[3]*, R. L. S. da Silva[4]

Post-Graduation Program on Computer Science, Federal University of Juiz de Fora, Rua José Lourenço Kelmer, S/n - Martelos
Juiz de Fora, Minas Gerais, 36036-330, Brazil
[1]*akio@ice.ufjf.br,* [2]*luciano.wxc@gmail.com,* [3]*marbervi@gmail.com,* [4]*rodrigoluis@gmail.com*

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

In the pairwise rigid registration problem, we need to find a rigid transformation that aligns two point clouds. The classical and most common solution is the Iterative Closest Point (ICP) algorithm. However, the ICP and many of its variants require that the point clouds are already coarsely aligned. We present in this paper a method named Shape-based Weighting Covariance Iterative Closest Point (SWC-ICP) which improves the possibility to correctly align two point clouds, regardless of the initial pose, even when they are only partially overlapped, or in the presence of noise and outliers. It benefits from the local geometry of the points, encoded in second-order orientation tensors, to provide a second correspondences set to the ICP. The cross-covariance matrix computed from this set is combined with the usual cross-covariance matrix, following a heuristic strategy. In order to compare our method with some recent approaches, we present a detailed evaluation protocol to rigid registration. Results show that the SWC-ICP is among the best compared methods, with a better performance in situations of wide angular displacement of noisy point clouds.

Keywords: Rigid Registration; Iterative Closest Point; Orientation Tensor; Shape Dissimilarity; Computational Geometry.

1. Introduction

Surface registration is a problem found in many areas, such as shape acquisition, medical images support, simultaneous localization and mapping (SLAM), quality inspection, and others. In this paper we focus on pairwise rigid registration, a subproblem where a rigid transformation is sought to align two point clouds.

The classical solution to rigid registration is the Iterative Closest Point (ICP)[1]. Issues like only partially overlapped scans, point clouds corrupted by noise and outliers, or incomplete models, due to self-occlusion or scan range limitation, restrain its applicability. Many variants[2] of the ICP tackle these issues. Like the original ICP

*Corresponding Author

they have an additional limitation that imposes that the point clouds are already coarsely aligned, i.e., the angular displacement between them is low, whether by the result of an algorithm or manually positioning them. Since there is not a definitive solution to rigid registration and this is a primary step for other applications, algorithms that achieve smaller residual errors in a broader range of situations are needed.

We propose a method called Shape-based Weighting Covariance ICP (SWC-ICP), an improvement over the original Iterative Closest Point (ICP) algorithm[1]. The local geometric information of the points is encoded in second-order orientation tensors. We use two cross-covariance matrices: one is computed from the correspondences of similar tensors, and the other is computed from the usual Euclidean correspondences. These two matrices are combined following a heuristic weighting strategy. Our method, unlike most variants of the classical approach, is able to improve the possibility to successfully align point clouds with wide angular displacement, even when corrupted by noise and outliers.

Several tests are performed using a proposed quantitative evaluation protocol, in order to give some statistical relevance to our results, when compared to recent approaches. This protocol includes the description of the generation process of the synthetic outliers, additive noise and partially overlapped point clouds. It also presents a scoring system to compare the results of each algorithm. We use a dataset composed by four point clouds with different topologies and geometries, high and low curvature points, holes and some degree of symmetry. This dataset is available at `www.gcg.ufjf.br/datasets/icp-models.zip`.

Our results show that the SWC-ICP is among the best methods evaluated in all point clouds tested, even in situations of noise and outliers. The main feature of our method is the consistency of the results, regardless of the initial angular displacement of the point clouds.

The main contributions of this work are twofold: a method that improves the possibility of correct convergence of the ICP in cases with wide angular displacement, and a detailed comparison protocol for rigid registration.

1.1. *Related works*

The survey written by Salvi *et al.*[3], proposes a taxonomy for rigid registration methods, classifying them in coarse and fine registration, highlighting the strengths and weaknesses of each. Coarse registration methods aim to give a good initial guess of the rigid transformation between two point sets. These methods usually try to be more robust to noise, outliers and wide initial displacement, at the cost of a lower accuracy. The survey of Diez *et al.*[4] further classify coarse methods according to a proposed pipeline, composed by three optimization stages: keypoints detection, description and search strategy.

Fine methods, on the other hand, assume that the point clouds are already coarsely aligned and aim to find the most accurate result as possible. Most of fine

methods are based on modifications of the ICP algorithm. Sharp *et al.*[5,6] propose the use of invariant features like second order moment, curvature or spherical harmonics to define a hybrid distance measure between points, weighted by a factor α, fixed or updated with the Mean Squared Error (MSE) on each iteration of the algorithm. Some authors attach local covariance matrices to the points, so that the local geometry is considered. The Generalized-ICP[7] and Multi-Channel Generalized-ICP[8] are some examples, with the latter using additional channels other than the position of the points to improve the ICP. The A-ICP[9] accounts the anisotropic localization error into a covariance matrix, representing a local zero-mean Gaussian distribution that is used in the ICP. The method also allows the use of different covariance matrices, according to the specifics of the problem.

Another line of improvements adopts the use of non-linear optimization strategies. Bouaziz *et al.*[10] use sparsity-inducing l_p-norms with an Alternating Direction Method of Multipliers (ADMM) optimizer. The norm parameter p controls the robustness to outliers, but has a computational time trade-off. As p gets closer to 0 the method becomes more robust to noise and outliers, however the time required is unfeasible. Mavridis *et al.*[11] relieves this problem adapting a Simulated Annealing process before the ADMM optimization. Yang *et al.*[12] propose the Globally Optimal ICP, the first method to achieve the global optimum under the norm l_2, according to the authors. The method uses a Branch-and-Bound scheme to search the 3D motion space SE3 for the best transformation.

Reyes *et al.*[13] presented a different approach, using geometric algebra to solve the rigid registration problem, that is not based on the ICP. Their method uses a tensor voting framework to find a plane representing the affine motion in the geometric algebra space. Tensor voting is a strong tool to find coplanar structures, thus allowing the authors to find these planes. Also, thanks to tensor voting, their method can cope with high amounts of outliers.

Even though the fine registration literature lists papers over twenty years old, the comparison protocol has not matured enough. There are many authors that do not fully present the behavior of their methods under different situations. The absence of benchmarks also makes hard the comparison between the methods. Noise, outliers, occlusion and partially overlapped point clouds are usual issues addressed. While these are valid scenarios to show their potential improvements, these papers lack a deeper statistical analysis. Time, residual error and number of iterations are the common metrics, but the relevance of the proposed method can not be guaranteed with just few examples that might be biased by the initial transformation applied. We understand that quantitative analyses are computationally expensive, specially with huge point clouds. To run several trials, with different parameters and compare with other algorithms requires too much computational time, even with current parallelism technologies.

Although this kind of analysis is rare, we highlight some examples found in the literature. Sharp *et al.*[5] ran 100 random transformations and presented a table

with the convergence rate to a location near the ground truth. Reyes *et al.*[13] did a similar experiment, but only with 20 trials, measuring the success rate. In their work an experiment is deemed a success when at least 50% of the correspondences were correctly identified. Jian *et al.*[14] measured the convergence range angles for 2D rigid registrations, and also success rates for 3D partially overlapped range images, with 30 different transformations. The random nature of the transformations used reflects better the performance of the methods in real scenarios.

This work is closely related to the ICP-CTSF[15], that enhances the matching step of the ICP using a strategy very similar to Sharp *et al.*[6], however, it implements a different invariant feature and the weighting factor is updated only when a local optimal solution is reached, rather than in each iteration. Our method use the same weighting strategy, but we modify the transformation estimation step instead.

The remaining part of this paper is structured as follows: Section 2 presents the original ICP, the classical solution for rigid registration that we propose to modify. Section 3 gives an overview of the algorithm used to encode the local geometry in second-order orientation tensors. The Section 4 presents our previous work, with the tensor dissimilarity factor in Section 4.1 and the base weighting strategy of the ICP-CTSF in Section 4.2. We present our method in Section 5. The Section 6 contains the description of the evaluation protocol, the experimental results and comparisons of our method with recent approaches. Finally, Section 7 concludes with a discussion about the proposed method.

2. Iterative closest point

The original ICP[1] requires only two point clouds, named the Model set $M = \{m_i \mid m_i = (m_{ix}, m_{iy}, m_{iz})\}$ and the Data set $D = \{d_j \mid d_j = (d_{jx}, d_{jy}, d_{jz})\}$.

Two major steps compose the ICP. The first step is the matching, which builds a correspondences set $E = \{e_j \mid e_j = N_E(d_j, M)\}$, where $N_E(d_j, M)$ is defined as the Euclidean closest point operator, that returns for each point d_j, the point m_i which minimizes $||d_j - m_i||_{l_2}$. The correspondences set and the Data set are passed to the next step.

In the second step, called the transformation estimation, the method tries to find the rotation $\mathbf{R}$ and the translation $\vec{\mathbf{t}}$ that best align the two sets received. The usual objective function is:

$$\min_{\mathbf{R},\vec{\mathbf{t}}} \left(\sum_{i=1}^{|D|} ||e_i - \mathbf{R} \cdot d_i - \vec{\mathbf{t}}||^2 \right). \tag{1}$$

The original ICP estimates transformations in $\mathbb{R}^3$ using the method of Unit Quaternions, a closed-form solution developed by Horn[16]. So in order to find the rotation that minimizes Eq. 1, the Unit Quaternions method needs to find a quater-

nion instead. First, the centroids of each set are calculated:

$$\mu_D = \frac{1}{|D|}\sum_{i=1}^{|D|} d_i, \quad \mu_E = \frac{1}{|E|}\sum_{i=1}^{|E|} e_i. \tag{2}$$

The cross-covariance Σ_{DE} between the Data set and the correspondences set is used to build the matrix L, that encodes the relationship between both sets. The cross-covariance matrix is given by:

$$\Sigma_{DE} = \frac{1}{|D|}\sum_{i=1}^{|D|}[d_i e_i^T] - \mu_D \mu_E^T. \tag{3}$$

$$L = \begin{bmatrix} \Sigma_{11}+\Sigma_{22}+\Sigma_{33} & \Sigma_{32}-\Sigma_{23} & \Sigma_{13}-\Sigma_{31} & \Sigma_{21}-\Sigma_{12} \\ \Sigma_{32}-\Sigma_{23} & \Sigma_{11}-\Sigma_{22}-\Sigma_{33} & \Sigma_{12}+\Sigma_{21} & \Sigma_{31}+\Sigma_{13} \\ \Sigma_{13}-\Sigma_{31} & \Sigma_{12}+\Sigma_{21} & -\Sigma_{11}+\Sigma_{22}-\Sigma_{33} & \Sigma_{23}+\Sigma_{32} \\ \Sigma_{21}-\Sigma_{12} & \Sigma_{31}+\Sigma_{13} & \Sigma_{23}+\Sigma_{32} & -\Sigma_{11}-\Sigma_{22}+\Sigma_{33} \end{bmatrix}. \tag{4}$$

The eigenvector associated to the greatest eigenvalue of L represents the direction of maximum correlation. This eigenvector is chosen as the optimal quaternion, whose associated rotation $\mathbf{R}$ minimizes Eq. 1. The optimal translation $\vec{\mathbf{t}}$ is obtained as the difference vector between the centroid of the correspondences set and the centroid of the rotated Data set:

$$\vec{\mathbf{t}} = \mu_E - \mathbf{R}\cdot\mu_D. \tag{5}$$

With $\mathbf{R}$ and $\vec{\mathbf{t}}$ computed, the transformation is applied to all points in the Data set. These two major steps are iterated until a stopping criterion is satisfied. An error threshold or number of iterations are the usual choices.

Besl and McKay assume that initially both Model and Data point clouds are coarsely aligned. As the matching and transformation estimation steps are iterated, the number of correct correspondences increases, leading to a correct alignment. However, this assumption is too strong for most cases, especially in the real world scenarios. If the method assigns too many wrong correspondences in the matching step, it is unable to estimate a relative pose better than the previous, which results in a failure case.

3. Local geometry estimation

The method we use to estimate the local geometry, presented by Cejnog[15], is very similar to the Tensor Voting framework[17],[18]. The geometric disposition of the neighborhood of each point is encoded in tensors. Each point casts its influence on the neighborhood through a vector voting field. Each vote vector is converted to a tensor and accumulated on the neighbors. Since the local geometry does not change under rigid transformations, it can be used as an invariant feature, suitable for rigid registration.

Cejnog[15] used in his method two different voting fields: one isotropic followed by one anisotropic. The first gives an initial approximation of the geometry, based

solely on the relative position between the points. The second field requires a previous estimation and must be applied after the isotropic field. It enhances the approximation of the geometry enforcing coplanar structures. A different distance metric is used in the second field, based on the distance over an elliptical trajectory. But unlike Cejnog, we use only the first voting field to give an approximation of the geometry. For rigid registration, the application of the first voting field by itself produces better results.

Since we are dealing with point clouds and do not have any previous information about the vicinity of the points, the neighborhood of a point p is represented by a list $L_k(p)$ of its k nearest-neighbors sorted by their Euclidean distances. The list size k is the main parameter of this preprocessing stage. For the following text, k is used as a percentage of total points in the point cloud.

3.1. *Isotropic voting field*

The isotropic voting field[15] builds for each point p, belonging to a point cloud P, a second-order tensor $\mathbf{T}_p$, which accumulates the weighted sum of tensors built from the vote vectors $\overrightarrow{pq}$, for each neighbor $q \in L_k(p)$.

The tensors $\mathbf{T}_{pq}$ formed by the product of the normalized $\widehat{pq} \cdot \widehat{pq}^T$, are accumulated. A Gaussian decay is used, proportional to the Euclidean distance between p and q with standard deviation σ_p. This deviation is such that the farthest neighbor q_f has influence 0.01:

$$\sigma_p = \sqrt{\frac{||\overrightarrow{pq_f}||^2}{\ln 0.01}}. \tag{6}$$

The output are the tensors $\mathbf{T}_p$:

$$\mathbf{T}_p = \sum_{q \in L_k(p)} e^{\frac{-||\overrightarrow{pq}||^2}{\sigma_p^2}} \cdot \widehat{pq} \cdot \widehat{pq}^T. \tag{7}$$

Since the method only deals with the relative position of the points, higher order elements like orientation and curvature sign are not identified. However, the tensors estimated are distinguishable enough to be used as an invariant feature in the matching step of the ICP.

4. Comparative Tensor Shape Factor

4.1. *Dissimilarity factor*

The Comparative Tensor Shape Factor (shortened CTSF)[15] can be used whenever the shape of two second-order orientation tensors should be compared. Such tensors can be represented by a hyper-ellipsoid, whose axes are proportional to the eigenvalues of the tensor matrix. Thus, to compare the eigenvalues between two tensors is equivalent to compare the shapes of the respective hyper-ellipsoids. Since what

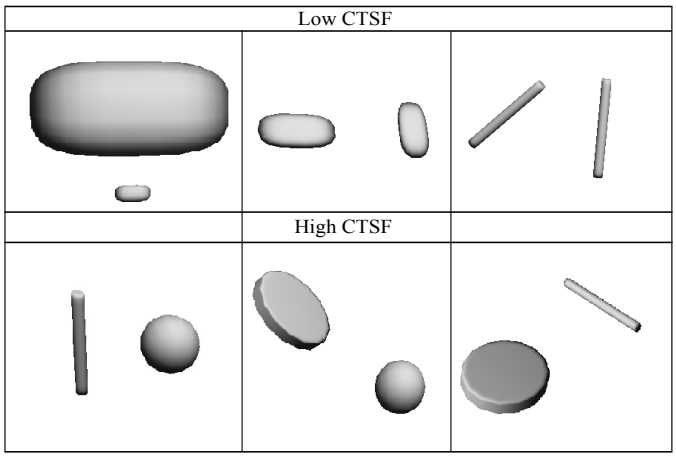


Fig. 1: Examples of the CTSF Between Two Tensors.

matters is the shape, the tensors used are normalized, i.e., $\sqrt{\sum_{i=1}^{N} \lambda_i^2} = 1$. The CTSF between two tensors $\mathbf{S}_1$ and $\mathbf{S}_2$ is:

$$CTSF\left(\mathbf{S}_1, \mathbf{S}_2\right) = \sum_{k=1}^{N} \left(\lambda_k^{\widehat{\mathbf{S}}_1} - \lambda_k^{\widehat{\mathbf{S}}_2}\right)^2, \tag{8}$$

where $\lambda_i^{\widehat{\mathbf{S}}_1}$ and $\lambda_i^{\widehat{\mathbf{S}}_2}$ are the ith eigenvalues of the normalized tensors $\widehat{\mathbf{S}}_1$ and $\widehat{\mathbf{S}}_2$, in a space with dimension N.

The CTSF is a factor of dissimilarity, with smaller values indicating more similar tensors. Since the shape of a hyper-ellipsoid is invariant to rigid transformations, the CTSF can be used to provide approximated correspondences. Figure 1 shows some examples of low and high CTSF between two tensors. Note that the scale and main directions of tensors do not affect the CTSF when their shapes are similar.

4.2. *Shape-based matching*

Following the premise that a better matching scheme leads to a better transformation, the ICP-CTSF[15] implements a matching strategy using the tensor shape of each point to improve the number of correct correspondences. The CTSF presented in Section 4.1 is used side by side with the Euclidean distance to produce the correspondences set.

The ICP-CTSF assumes that the point corresponding to the same region in two different point clouds will have very similar tensors after the preprocessing stage, since its surrounding geometry should be the same. Thus, the CTSF between them should be very low. So no matter how apart a point cloud is from the other, the shape-based matching tends to correlate points whose vicinity have similar shapes.

Since the shape of the tensors does not change during rigid registration, the shape-based correspondences are always the same, and using only them in the ICP would make it converge to a local optimal solution very fast, often just coarsely aligned. During the iterative process, however, the Euclidean correspondences change and the match is improved as the point clouds are getting aligned.

The formulation of the matching used in the ICP-CTSF is:

$$d_{CTSF}(d_i, m_j) = ||d_i - m_j||_2 + w_n \cdot CTSF\left(\mathbf{S}_{d_i}, \mathbf{S}_{m_j}\right),$$
$$w_n = w_0 b^n, b < 1 \text{ and } 0 \leq w_n < w_0,$$

where n is incremented whenever a local optimal solution is reached, the parameter w_0 is the initial weight given to the CTSF and b controls the variation rate of the weighting factor towards zero. The ICP-CTSF stops when $w_n \approx 0$ and it is unable to improve the RMS error between the two point clouds.

The ICP-CTSF combine the correspondences found using the CTSF and the Euclidean matching. In order to align point clouds with large angular displacement, the shape-based matching must have more relevance in the first iterations, that is, w_0 should be a high value. When a local optimal solution is reached, the current pose is hopefully better than the initial, and the Euclidean matching should produce better correspondences. The method then reduces the influence of the CTSF. Throughout the iterations, the weight of the CTSF tends to zero, and the method becomes more similar to the original ICP. The first iterations of the method coarsely align the point clouds, and the latter iterations perform fine alignment.

5. Proposed method

Our method follows the same coarse-to-fine strategy of the ICP-CTSF, that gives more importance to the shape-based correspondences in the beginning of the registration, and more importance to the Euclidean correspondences in the end.

We apply this strategy directly in the ICP transformation estimation step, using two cross-covariance matrices. One established by the Euclidean distance correspondences, as usual, and another by the correspondences of points whose tensor shapes are similar, i.e, points with low CTSF.

The cross-covariance matrix using Euclidean correspondences is computed exactly in the same way of the original ICP. The second cross-covariance uses a different correspondences set, defined as $S = \{s_i \mid s_i = N_{CTSF}(d_i, M)\}$, with $N_{CTSF}(d_i, M)$ as the operator that returns the point in M whose tensor shape is the most similar to the tensor shape of d_i, according to the CTSF of both. Like in Eq. 2, the centroid of this shape-based correspondences set is:

$$\mu_S = \frac{1}{|S|} \sum_{i=1}^{|S|} s_i. \tag{9}$$

If we minimize only the distance between points whose shapes are similar, the

cross-covariance matrix would be expressed by:

$$\Sigma_{DS} = \frac{1}{|D|}\sum_{i=1}^{|D|}[d_i s_i^T] - \mu_D \mu_S^T. \tag{10}$$

Consider the sum of both cross-covariance matrices from Eq. 3 and 10, weighted by a factor w_n:

$$\begin{aligned}\Sigma_{DES} &= \tfrac{1}{|D|}\sum_{i=1}^{|D|}[d_i e_i^T + w_n d_i s_i^T] - [\mu_D \mu_E^T + w_n \mu_D \mu_S^T] \\ &= \tfrac{1}{|D|}\sum_{i=1}^{|D|}[d_i (e_i^T + w_n s_i^T)] - [\mu_D (\mu_E^T + w_n \mu_S^T)],\end{aligned} \tag{11}$$

where $w_n = w_0 b^n,\ b < 1$ and $0 \leq w_n < w_0$, n is incremented whenever a local optimal solution is reached. The parameter w_0 is the initial weight given to the shape-based correspondences, and b controls the update size of the weighting factor. This weighting strategy is the same presented in Section 4.2.

The cross-covariance matrix Σ_{DE} (Eq. 3), represents the multivariate dispersion of Euclidean points correlated using the nearest neighbors strategy as proposed originally[1]. The cross-covariance matrix Σ_{DS}, in its turn, represents the multivariate dispersion of Euclidean points correlated by using the lowest CTSF.

Both matrices are symmetric, positive semi-definite, and thus admit a linear combination which results in a matrix whose eigensystem represents the contributions of both dispersions, provided that their independent variables are of the same kind, i.e., 3D coordinates. This is exactly what is used in tensor voting approaches[17].

Therefore, we can further expand Eq. 11 to use more cross-covariance matrices, each obtained with f different matching operators:

$$\begin{aligned}\Sigma = \frac{1}{|D|}\sum_{i=1}^{|D|} &[\, d_i (w_{n_1} e_i^T + w_{n_2} s1_i^T + w_{n_3} s2_i^T + \cdots + w_{n_f+1} sf_i^T)] - \\ &[\, \mu_D (w_{n_1} \mu_E^T + w_{n_2} \mu_{S1}^T + w_{n_3} \mu_{S2}^T + \cdots + w_{n_f+1} \mu_{Sf}^T)],\end{aligned} \tag{12}$$

where w_{n_o} is the relative weight to the n-th local optimal solution of the correspondence of the o-th matching operator, and $w_{n_o} = w_0 b^n, b < 1$ and $0 \leq w_n < w_0$. In our case, $f = 2$, $w_{n_1} = 1$ and w_{n_2} is variable. This linear combination of cross-covariance matrices is one of the main contribution of our work.

The matrix L of Eq. 4 is computed as usual, using Σ_{DES}, instead of Σ_{DE} from Eq. 3. The rotation and the translation are also computed exactly in the same way of the original ICP.

Analyzing the formulation of Eq. 11, we note that instead of using the closest point or the point whose shape of the tensor is the most similar, we use a linear combination of both. It means that the considered correspondence is not an existing point, but after a large number of local optimal solutions, it tends to be the closest point, since the weighting factor approximates to zero. Results show that

this approach has a higher rate of convergence to the correct solution than some recent methods when the point clouds have a wide angular displacement.

6. Experimental results

6.1. *3D model dataset setup*

In order to create an experimental setup able to perform a fair comparison of methods, we define a trial as an execution of rigid registration under a certain amount of noise, outliers and overlapping, with an angular displacement. These are common issues found in the literature. Our test suite ensures that the same trial is performed by the different compared algorithms.

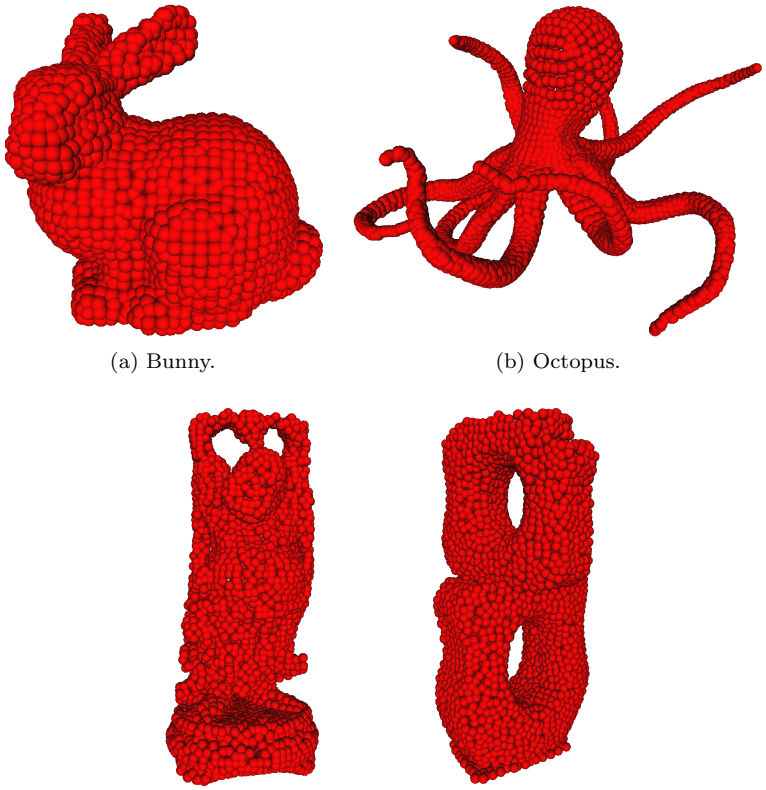
For practical applications the point clouds are assumed to be two distinct representations from the same object or complementary views from a scene. However, for performance analysis and comparisons with other methods synthetic point clouds are often used, allowing varying degrees of overlapping, noise and outliers. In these controlled cases, the overlapped regions are identical, and all points within it, have an exact correspondence.

We use four base point clouds: Bunny and Happy Buddha, provided by Stanford University Computer Graphics Laboratory, Octopus, provided courtesy of INRIA by the AIM@SHAPE-VISIONAIR Shape Repository, and Genus-2, provided by École Polytechnique Fédérale de Lausanne Computer Graphics and Geometry Laboratory. The Figure 2 shows the visual representation of the point clouds. They have different features, such as holes, some degree of symmetry and points with high and low curvatures. We sampled the point clouds to make possible the execution of multiple trials in a feasible time. The Bunny model used is the smallest zippered version available. The other point clouds were sampled using a Poisson-disk sampling algorithm[19]. All point clouds used are normalized with the greatest edge of the bounding box scaled to have size 1.

We divide our tests in two groups, total overlapping and partial overlapping, running 30 trials for each configuration case. In the first group, we experiment adding different amounts of noise, outliers and both at the same time. In the second group we evaluate how the size of the overlapping and individual regions affects the quality of registration.

A rigid transformation is applied in the Data set to simulate the initial state of the point clouds. The rotation axis is set as a random normalized isotropic vector. The range of angles used varies from 15° to 180°, sampled each 15°. Therefore, twelve angles are used.

Outliers are generated using a uniform distribution over a sphere with radius twice the size of the greatest edge of the bounding box. We distribute the outliers in a sphere bounding the Model and Data sets because the constant curvature of their synthetic external surfaces will not significantly take part in the shape alignment. Noise is simulated adding to each point p_i a random normalized isotropic vector $\vec{r}$. The magnitude of this vector is obtained using a Gaussian random variable weighted



(a) Bunny. (b) Octopus.

(c) Happy Buddha. (d) Genus-2.

Fig. 2: The base point clouds used on the experiment: (a): Bunny (containing 1889 points). (b): Octopus (3822 points). (c): Happy Buddha (3118 points). (d): Genus-2 (2711 points).

by a scale δ that controls the intensity of the noise:

$$p_i = p_i + \delta \cdot \mathcal{N}(0,1) \cdot \vec{r}. \tag{13}$$

Partially overlapped point clouds are generated using a region growing algorithm, using the list of closest neighbors $L_k(p)$. First, the common region is determined from a random point as initial seed. This region grows until it has a number of points β. The α remaining points of the unique region of each point cloud are obtained in the same way, using as initial seed a random point next to the border of the overlapping region.

The values for noise used are: $\delta = \{0.01, 0.05\}$. Higher amounts of noise are not considered because the point cloud loses most of its features and such level of degradation is an unrealistic case. Outliers are tested with $5\%, 20\%, 50\%$ of the

Table 1: Percentage of Points of the Individual Region (α) and of the Overlapping Region (β) for the Second Group of trials.

α	12.5%			25%		
β	75%	50%	25%	50%	37.5%	25%

number of points, a case with a small amount, moderated amount and a very corrupted point cloud, respectively. The trials with noise and outliers together use all the combinations of these values.

Each trial use different seeds to avoid biased generation of noise, outliers and overlapping regions. These seeds are generated by a Mersenne Twister[20] pseudo-random number generator.

In the second group of experiments we evaluate different combinations of overlapping (β) and individual region (α) sizes. Table 1 shows the combinations used. The first case with $\alpha = 12.5\%$ and $\beta = 75\%$ means: "Each point cloud has 12.5% unique original points and they mutually share 75% as common overlapped points." Note that the amount of individual region points α is always the same for both point clouds. In this way, with $\alpha = 25\%$ the range of overlapping is shortened, since it is not possible to have more than 50% of overlapped points and 25% of remaining points.

6.2. *Evaluation protocol*

In this work we use the GTRMS error (Ground Truth Root Mean Squared error)[15], the root mean squared distance between all points of the Data set and their ground truth correspondences in the Model set. This is only possible because we are using synthetic pairs of point clouds, and we know in advance the correct correspondences. This error is measured only for inliers and points in the overlapped region, because only these points have a correct correspondence. We use this variation of error because we are interested in measuring how close to the correct registration the methods are, instead of how close one point cloud is to the other. Note that this error is used only for evaluation purposes, and the error used inside the SWC-ICP is the usual RMS.

Another measurement we adopted is the Labeled Error[15], the number of correct correspondences found by the original matching function of the ICP. This measure is also only possible using synthetic pairs of point clouds. It is an indicative of how close the regions of interest are.

Our previous work[15] used a threshold on the GTRMS and Labeled Error to define a success case. However, the trials frequently have intermediate values, for registrations that have success on the coarse stage, but failed in the fine stage. If we take just the mean error with such trials, we might not express the correct performance of the method, and induce a wrong analysis. The same happens to

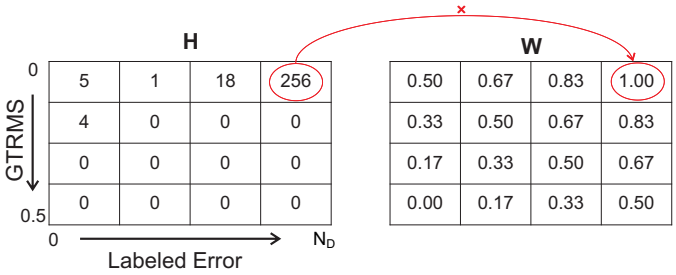


Fig. 3: Example of a 4×4 bins histogram matrix $\mathbf{H}$ and a 4×4 weighting matrix $\mathbf{W}$. In this example the Score is: $\frac{1}{284}((5 \times 0.50) + (1 \times 0.67) + (\cdots)) = 0.9698$.

the Labeled Error. To overcome this issue, we propose a scoring system combining these two measurements to compare the performance of the methods.

First a 2D Histogram $\mathbf{H}$ is computed using the GTRMS and Labeled Error from all trials of a given parameter combination. We build this histogram with 40×40 bins. The error ranges used are $[0.0, 0.5]$ for the GTRMS, and $[0, N_D]$ for the Labeled Error, where N_D is the number of points in the overlapped region of the point clouds tested. The score is a weighted mean of the histogram. We use a weighting matrix $\mathbf{W}$ with the same dimensions of the histogram and it is obtained assigning larger values for the cells with small GTRMS and high Labeled Error, and conversely, smaller values for cells with high GTRMS and small Labeled Error. The remaining cells are filled following a bilinear interpolation. Figure 3 is an example of score for a trial, showing a 4×4 bins histogram and its weighting matrix.

$$Score = \frac{\sum_{i=1}^{N_{bins}} \sum_{j=1}^{N_{bins}} \mathbf{H}_{ij} \times \mathbf{W}_{ij}}{\sum_{i=1}^{N_{bins}} \sum_{j=1}^{N_{bins}} \mathbf{H}_{ij}}. \tag{14}$$

The score is normalized in the interval $[0, 1]$. However, the score is not a success rate. It is meant to be used to compare two methods in the same configuration of noise, outliers and overlapping amount. The labeled error is expected to decrease when noise is applied to the point cloud, since it is based on the number of correct correspondences found by the original ICP matching function. Therefore, the score tends to be smaller in such cases, but it does not mean the performance of the method is worse. Figure 4 summarizes the execution of our evaluation protocol.

6.3. *Results*

We compare the SWC-ICP with the original ICP and some recent algorithms to rigid registration:

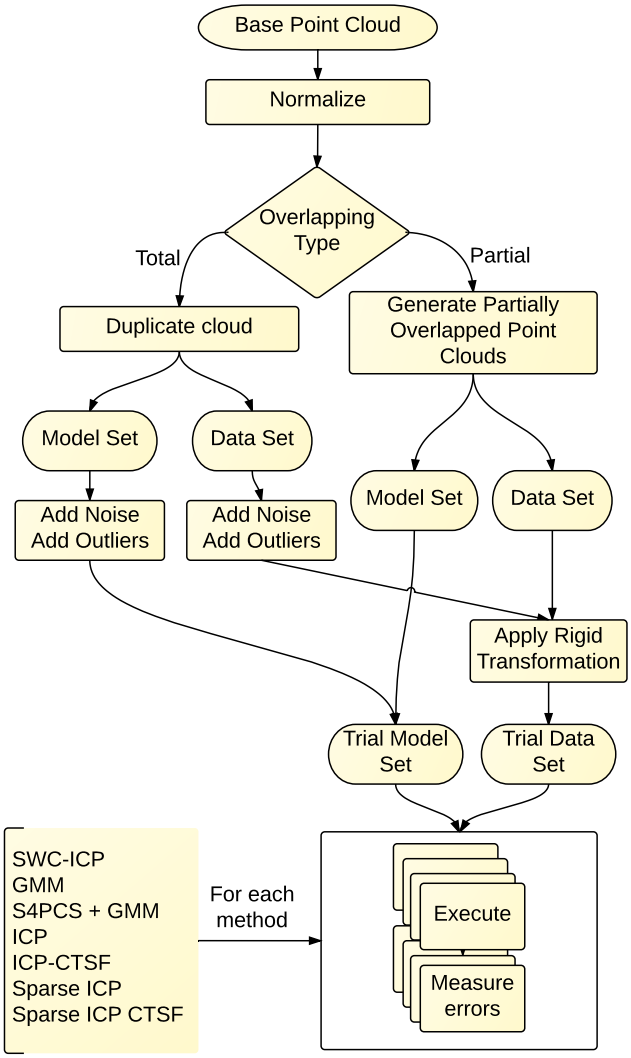


Fig. 4: Flow chart of our evaluation protocol.

- The GMM framework[14], that represents the point clouds as Gaussian Mixture Models and reduce the rigid registration to the problem of minimize the statistical discrepancy between two mixtures.
- The Super 4PCS (S4PCS)[21], an improvement of the 4PCS, a coarse alignment method with good results, comparable to some fine methods. It is

based on the RANSAC, but uses an improved search strategy to find congruent 4-point bases between the point clouds. To make comparisons with our method fair, we run the Super 4PCS followed by the GMM, to exemplify the behavior of a coarse method followed by a fine method.
- The Sparse ICP[10], that uses l_p-norms with an Alternating Direction Method of Multipliers optimizer. The parameter p controls the robustness to noise and outliers, and is a trade-off between accuracy and computational time.
- The ICP-CTSF[15], our previous method, that uses the CTSF to estimate the best corresponding neighbors in the matching step of the ICP.
- We also compare with a variation of the Sparse ICP with the CTSF[15], that uses the shape-based matching to find the correspondences and minimizes using the ADMM, like the original Sparse ICP.

We implemented the original ICP, the ICP-CTSF and the modification of the Sparse ICP to adequate the CTSF matching, while keeping the core method the same as the provided Sparse ICP. All other methods used codes made available by the authors. All codes are written in C/C++.

6.3.1. *Results with noise and outliers*

We evaluate six different sizes of the neighbors list of the SWC-ICP: 100%, 75%, 50%, 15%, 5%, 1%. The update size of the weighting factor used was $b = 0.1$. It is an intermediate value that does not take too many updates, and neither finishes the update too soon, without proper exploration of the search space. The S4PCS was set with: $\delta = 0.005$, no filtering by angle, normals, distance or color, and no further sampling of the point cloud. The Sparse ICP and Sparse ICP with CTSF were set with parameters: $p = 0.4$, $\mu = 10.0$, $\alpha = 1.2$, $max_\mu = 10^5$, $max_{icp} = 100$, $max_{outer} = 100$, $max_{inner} = 1$, $stop = 10^{-4}$. Since the ICP-CTSF and the Sparse ICP with CTSF use tensors to match points like our method, using only the first voting field have an effect on their results. Therefore, we present results using only the isotropic voting field, differently from the results presented by Cejnog[15]. These two methods use $k = 50\%$, since it is the neighbors list size that produces the best results. All the other methods use the best configuration presented by the authors and the default when not specified.

Tables 2, 3, 4 and 5, present the score obtained by each method combining all twelve angles with the clean point cloud, with all situations with only outliers, all situations with only noise, all situations with noise and outliers together, and the overall score. This overall value is our final score for the methods compared in the total overlapping test.

Figures 5 and 6 show the score combining the clean point cloud, outliers, noise and outliers together with noise, this time detailed by angle. The graphics show the best SWC-ICP, the GMM, the Super 4PCS with GMM, the original ICP, the

Table 2: Combined score for each method on the Bunny.

Bunny					
Method Name	Clean	Outliers	Noise	Noise + outliers	Overall
SWC-ICP $k = 100\%$	1.0000	0.9340	0.7311	0.6547	0.7660
SWC-ICP $k = 75\%$	1.0000	0.9472	0.7308	0.6709	0.7774
SWC-ICP $k = 50\%$	1.0000	0.9389	0.7293	0.6614	0.7703
SWC-ICP $k = 15\%$	1.0000	0.9165	0.7226	0.6122	0.7390
SWC-ICP $k = 5\%$	1.0000	0.8861	0.6978	0.5173	0.6798
SWC-ICP $k = 1\%$	1.0000	0.9358	0.3295	0.1148	0.4297
GMM	0.5370	0.5261	0.3985	0.3859	0.4356
S4PCS + GMM	1.0000	0.4120	0.6972	0.2832	0.4441
ICP	0.3446	0.2365	0.2346	0.1747	0.2143
ICP-CTSF	1.0000	0.9774	0.7337	0.7002	0.8001
Sparse ICP	0.4610	0.3479	0.3260	0.2554	0.3074
Sparse ICP CTSF	1.0000	0.9898	0.7243	0.7152	0.8091

Table 3: Combined score for each method on the Octopus.

Octopus					
Method Name	Clean	Outliers	Noise	Noise + outliers	Overall
SWC-ICP $k = 100\%$	1.0000	0.7470	0.5767	0.4676	0.6000
SWC-ICP $k = 75\%$	1.0000	0.8476	0.5762	0.5135	0.6480
SWC-ICP $k = 50\%$	1.0000	0.8539	0.5757	0.5152	0.6503
SWC-ICP $k = 15\%$	1.0000	0.8257	0.5731	0.4532	0.6119
SWC-ICP $k = 5\%$	1.0000	0.7538	0.5626	0.3806	0.5559
SWC-ICP $k = 1\%$	1.0000	0.8602	0.3023	0.1731	0.4353
GMM	0.3793	0.3756	0.2519	0.2432	0.2891
S4PCS + GMM	0.9415	0.4903	0.5624	0.3070	0.4483
ICP	0.2817	0.2491	0.1967	0.1881	0.2126
ICP-CTSF	1.0000	0.9047	0.5779	0.4949	0.6533
Sparse ICP	0.3564	0.3232	0.2308	0.2210	0.2595
Sparse ICP CTSF	1.0000	0.9850	0.5352	0.5280	0.6828

ICP-CTSF, the original Sparse ICP and the Sparse ICP with CTSF.

Results with clean point cloud show that methods using the CTSF and the Super 4PCS were able to correctly register the point cloud in almost every case. With only outliers the methods using CTSF were better than the others. The tensor estimation algorithm is known for its robustness to outliers, confirmed with these results. Even with high amount of outliers, it was able to find the structure of the inliers, producing similar tensors on both model and data point clouds.

The scores of the case with only noise are lower than those with only outliers. This behavior is expected, since noise affects directly the original ICP matching function, used to count the labeled error. Therefore, this lower score does not mean that the methods failed more often than in cases with just outliers. The SWC-ICP with small k had poor results, with $k = 1\%$ being the worst result. A neighborhood with size 1% in the Bunny, for example, have only 19 points, which is too few to

Table 4: Combined score for each method on the Happy Buddha.

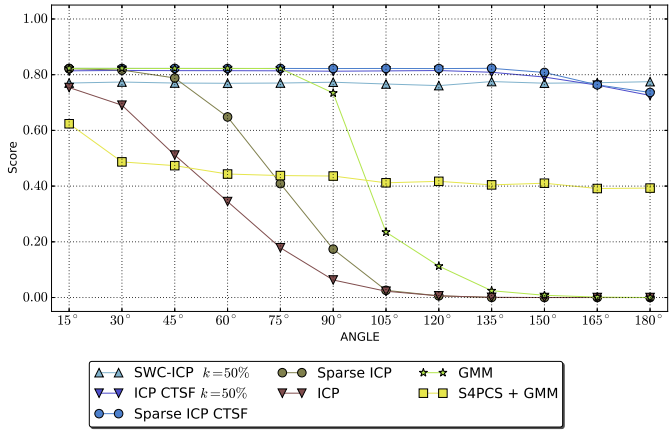
Happy Buddha					
Method Name	Clean	Outliers	Noise	Noise + outliers	Overall
SWC-ICP $k = 100\%$	1.0000	0.8100	0.6253	0.4698	0.6249
SWC-ICP $k = 75\%$	1.0000	0.9342	0.6353	0.5305	0.6880
SWC-ICP $k = 50\%$	1.0000	0.9533	0.6367	0.5631	0.7094
SWC-ICP $k = 15\%$	1.0000	0.9028	0.6097	0.4652	0.6433
SWC-ICP $k = 5\%$	1.0000	0.8797	0.5658	0.3930	0.5940
SWC-ICP $k = 1\%$	1.0000	0.9699	0.3053	0.1896	0.4715
GMM	0.5251	0.5195	0.3454	0.3510	0.4067
S4PCS + GMM	0.9977	0.6128	0.6239	0.3888	0.5353
ICP	0.4752	0.4155	0.3067	0.2953	0.3423
ICP-CTSF	1.0000	0.9512	0.6417	0.5792	0.7177
Sparse ICP	0.5385	0.4726	0.3548	0.3321	0.3882
Sparse ICP CTSF	1.0000	0.9350	0.6174	0.5685	0.7043

Table 5: Combined score for each method on the Genus-2.

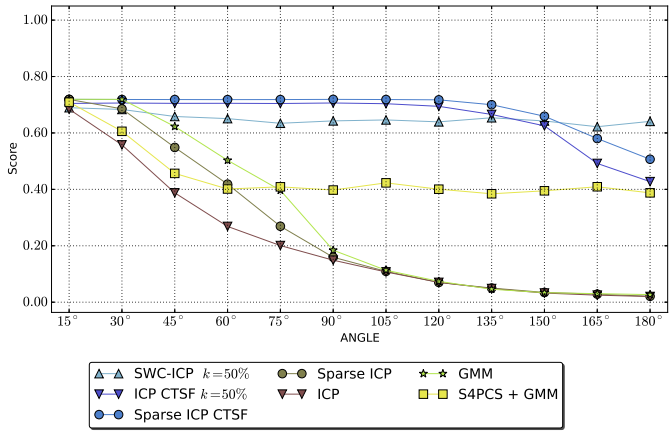
Genus-2					
Method Name	Clean	Outliers	Noise	Noise + outliers	Overall
SWC-ICP $k = 100\%$	1.0000	0.6281	0.5960	0.3654	0.5227
SWC-ICP $k = 75\%$	1.0000	0.8368	0.6292	0.4425	0.6189
SWC-ICP $k = 50\%$	1.0000	0.8845	0.6472	0.4864	0.6557
SWC-ICP $k = 15\%$	1.0000	0.8396	0.5891	0.3986	0.5909
SWC-ICP $k = 5\%$	1.0000	0.8473	0.4897	0.3383	0.5460
SWC-ICP $k = 1\%$	1.0000	0.9747	0.2441	0.1411	0.4387
GMM	0.5283	0.5383	0.3643	0.3594	0.4190
S4PCS + GMM	0.9790	0.4651	0.5335	0.2934	0.4335
ICP	0.4124	0.3842	0.2956	0.2677	0.3136
ICP-CTSF	1.0000	0.8686	0.6495	0.5075	0.6626
Sparse ICP	0.5359	0.4389	0.3659	0.3074	0.3691
Sparse ICP CTSF	1.0000	0.7775	0.5144	0.4611	0.5940

estimate a structure when corrupted by noise. In this way, the tensors obtained for a point tend to be too different between the model and the data point clouds, leading to bad matches. When noise and outliers are combined the results are similar to the case with just noise. The SWC-ICP with $k = 1\%$ is also the worst method, and the reason is the same, aggravated by the outliers.

Figures 5 and 6 show that although the SWC-ICP has a slightly smaller score than the ICP-CTSF, it was the best method in wide angles scenarios, with a very competitive score in the Bunny. The SWC-ICP basically is not affected by angle variations, as we can see almost a straight line in the graphics, for all the four point clouds. A similar behavior is observed in the Super 4PCS, that was designed as a coarse method, and thus should also not be affected by wide angle displacements. The scores of the ICP-CTSF and the Sparse ICP with CTSF decay at some point. What balances their result with the SWC-ICP is that before the decay, they have



(a) Bunny.

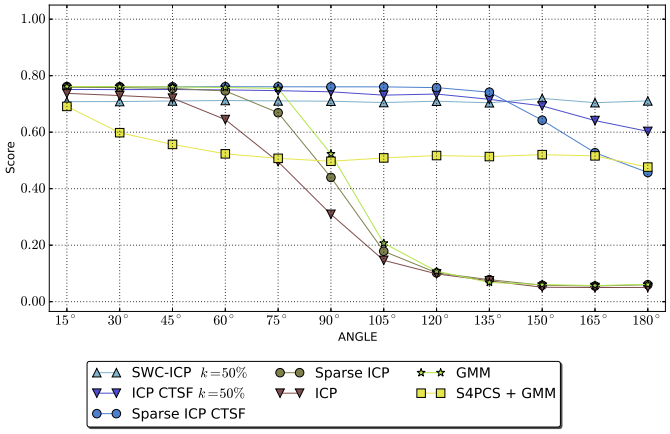


(b) Octopus.

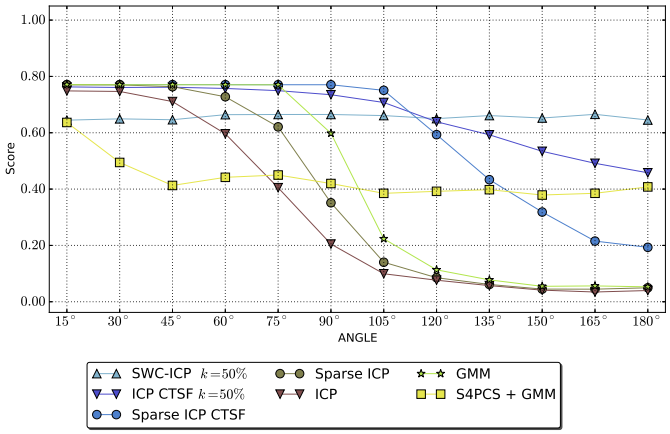
Fig. 5: Overall score per angle in the case with noise and outliers on the Buuny and Octopus.

a slightly better average score. The GMM framework, the Sparse ICP and the original ICP are not coarse methods, therefore the range of its good performance is limited by the angle. We can see in the graphics that these three methods even have competitive results with low angles, where they were supposed to perform better. In the Bunny, the Happy and the Genus-2, the GMM framework was the best fine-only method we experienced, with consistent results until 75°.

These results show that the SWC-ICP and the ICP-CTSF have different strengths and weaknesses, each better than the other at some point. The SWC-ICP is the best method for wider angle situations. The addition of the CTSF to



(a) Happy Buddha.



(b) Genus-2.

Fig. 6: Overall score per angle in the case with noise and outliers on the Happy Buddha and Genus-2.

the Sparse ICP also proved to have very good results, competitive to both the SWC-ICP and the ICP-CTSF.

6.3.2. *Results with partial overlapping*

In partial overlapping tests the Sparse ICP using the CTSF and the ICP-CTSF were run with $k = 1\%$, since it was the best k-neighborhood reported by Cejnog[15]. Four different neighbors list sizes were evaluated, the two greatest and two smallest sizes of the previous test. All other methods use the same parameters as the previous test. The SWC-ICP, the original ICP, the ICP-CTSF and both S4PCS use a trimmed

approach, discarding the 10% worse correspondences. We do not use the overlap amount of the test because in a real situation this value is unknown.

Table 6: Partial Overlapping Results on the Bunny

Bunny							
α	**12.5**%			**25.0**%			Overall
β	**25.0**%	**50.0**%	**75.0**%	**25.0**%	**37.5**%	**50.0**%	
SWC-ICP $k = 100\%$	0.1719	0.2858	0.6557	0.0679	0.0784	0.1797	0.2399
SWC-ICP $k = 75\%$	0.2353	0.3612	0.1074	0.0938	0.1365	0.1964	0.1884
SWC-ICP $k = 5\%$	0.4043	0.5115	0.8158	0.3356	0.4153	0.4928	0.5008
SWC-ICP $k = 1\%$	0.4067	0.5160	0.8162	0.3506	0.4217	0.4906	0.5003
GMM	0.2203	0.3349	0.4566	0.1297	0.1803	0.2611	0.2638
S4PCS + GMM	0.3420	0.6420	0.8689	0.1446	0.3191	0.5278	0.4741
ICP	0.1906	0.2223	0.2509	0.1512	0.1578	0.1778	0.1918
ICP-CTSF	0.4149	0.5340	0.8486	0.3459	0.4197	0.5004	0.5105
Sparse ICP	0.2550	0.3032	0.3692	0.2031	0.2490	0.2912	0.2784
Sparse ICP CTSF	0.5780	0.7564	0.8846	0.5498	0.6775	0.7493	0.6993

Table 7: Partial Overlapping Results on the Octopus

Octopus							
α	**12.5**%			**25.0**%			Overall
β	**25.0**%	**50.0**%	**75.0**%	**25.0**%	**37.5**%	**50.0**%	
SWC-ICP $k = 100\%$	0.2992	0.2603	0.3086	0.2240	0.2231	0.2056	0.2535
SWC-ICP $k = 75\%$	0.3188	0.3653	0.4457	0.2434	0.2241	0.2166	0.3023
SWC-ICP $k = 5\%$	0.4230	0.4548	0.5026	0.3389	0.3434	0.3874	0.4084
SWC-ICP $k = 1\%$	0.4320	0.4669	0.5006	0.3556	0.3637	0.4213	0.4234
GMM	0.3164	0.3397	0.3471	0.2721	0.2885	0.2719	0.3060
S4PCS + GMM	0.3993	0.4146	0.5831	0.2701	0.2637	0.2943	0.3783
ICP	0.2870	0.2934	0.2864	0.2504	0.2540	0.2438	0.2692
ICP-CTSF	0.4279	0.4632	0.5009	0.3487	0.3649	0.4204	0.4210
Sparse ICP	0.2658	0.3504	0.3393	0.2892	0.3370	0.3087	0.3151
Sparse ICP CTSF	0.4023	0.6796	0.8791	0.5001	0.6334	0.7052	0.6332

The Tables 6, 7, 8 and 9 show the scores detailed for all combinations of overlapping and individual regions presented on Table 1, combining the scores of all twelve rotation angles. The amount of overlap β and the amount of individual region α impact directly on the performance of all methods. The greater the overlapping and the smaller the individual region, the better are the scores. This behavior is expected, since the number of points with correct correspondences is larger.

The parameter k of the SWC-ICP have the opposite overall performance of the cases with noise and outliers. In this test the small values have better scores on all point clouds. With larger values of k, the tensors close to the edge of the overlapping region of the model point cloud get too different from the homologous tensors on the data point cloud, because the neighborhood in these cases may include points outside the overlapping region, exclusive to only one of the point clouds. Figure 7

Table 8: Partial Overlapping Results on the Happy Buddha

Happy Buddha							
α	**12.5**%			**25.0**%			Overall
β	**25.0**%	**50.0**%	**75.0**%	**25.0**%	**37.5**%	**50.0**%	
SWC-ICP $k = 100\%$	0.2959	0.3179	0.6900	0.2161	0.2598	0.3281	0.3513
SWC-ICP $k = 75\%$	0.2917	0.3731	0.7071	0.2309	0.2021	0.2742	0.3465
SWC-ICP $k = 5\%$	0.4104	0.4745	0.7121	0.3177	0.3855	0.4724	0.4621
SWC-ICP $k = 1\%$	0.4142	0.4793	0.7155	0.3359	0.3880	0.4722	0.4675
GMM	0.3122	0.2977	0.3489	0.2604	0.2848	0.2906	0.2993
S4PCS + GMM	0.3677	0.3968	0.5612	0.2789	0.2829	0.3568	0.3741
ICP	0.3003	0.3206	0.3489	0.2206	0.2393	0.2649	0.2824
ICP-CTSF	0.4189	0.4786	0.7498	0.3352	0.3882	0.4746	0.4742
Sparse ICP	0.3198	0.3955	0.4719	0.2567	0.3174	0.3725	0.3556
Sparse ICP CTSF	0.5014	0.7466	0.8748	0.5148	0.6488	0.7384	0.6708

Table 9: Partial Overlapping Results on the Genus-2

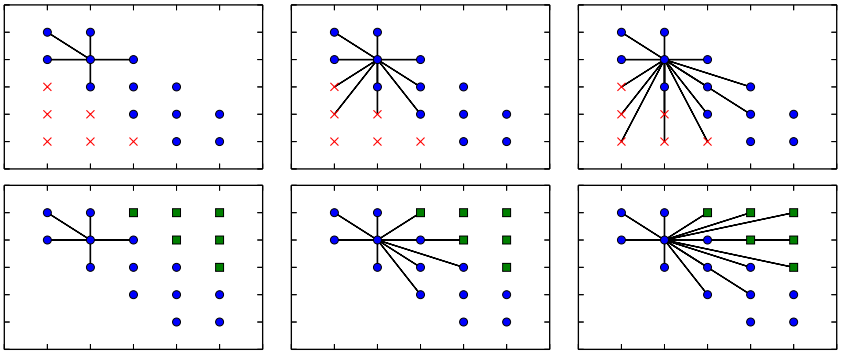
Genus-2							
α	**12.5**%			**25.0**%			Overall
β	**25.0**%	**50.0**%	**75.0**%	**25.0**%	**37.5**%	**50.0**%	
SWC-ICP $k = 100\%$	0.2449	0.3353	0.3374	0.2020	0.2395	0.1372	0.2494
SWC-ICP $k = 75\%$	0.2458	0.4096	0.4822	0.1798	0.2044	0.1361	0.2763
SWC-ICP $k = 5\%$	0.4333	0.5202	0.5672	0.3873	0.3880	0.4374	0.4556
SWC-ICP $k = 1\%$	0.4423	0.5223	0.5808	0.3903	0.3903	0.4381	0.4607
GMM	0.2985	0.3458	0.4238	0.2527	0.2787	0.2951	0.3158
S4PCS + GMM	0.3433	0.4313	0.5581	0.2506	0.2708	0.3336	0.3646
ICP	0.2937	0.3262	0.3472	0.2409	0.2320	0.2496	0.2816
ICP-CTSF	0.4426	0.5322	0.6729	0.3832	0.3706	0.4194	0.4702
Sparse ICP	0.3247	0.3999	0.4694	0.2933	0.3400	0.3975	0.3708
Sparse ICP CTSF	0.5177	0.7419	0.8318	0.5431	0.6667	0.7144	0.6693

shows an example of how the neighborhood gets different in points close to the edge of the overlapping region, as the number of neighbors increase.

In this way, $k = 1\%$ is the best neighborhood size for the SWC-ICP with partial overlapping point clouds. Some overlapping combinations are hard scenarios for all the methods, with scores frequently below 0.7. With a small overlapping region like 25%, the scores were even lower than 0.5. It happens because the methods failed in most cases to correctly align the point clouds, and in cases of failure the score is expected to be low.

The use of the CTSF on the Sparse ICP is very good for partial overlapping, achieving the best overall scores in all point clouds. The formulation of the Sparse ICP allows the method to succeed in partial overlapping scenarios, boosted by the better matching scheme using the CTSF.

In general, the SWC-ICP proved to be competitive in most cases, extending the range of convergence and achieving results as good as the best methods evaluated, with a superior performance in cases of wide angular displacement.



(a) 5 neighbors. (b) 10 neighbors. (c) 15 neighbors.

Fig. 7: Examples of nearest neighbors for a given point in a case of partial overlapping. Blue points are inside the overlapping region, while red and green are in the unique region of each point cloud. Note that as the number of neighbors increase, the neighborhood considers points outside the overlapping region, which yields different tensors.

7. Conclusion

We presented in this work the SWC-ICP, an improvement of the Iterative Closest Point algorithm. It benefits from the local geometry surrounding the points, encoded in second-order orientation tensors. The method improves the performance of the ICP, even in the presence of noise, outliers and partial overlapping, regardless of the initial pose. The main parameter is the size of the list of neighbors k, used to estimate the tensors. A suitable choice of k can filter out either local or global symmetries that can lead to mismatches. Similarly as occurs in multiresolution processes, a higher number of neighbors tend to reduce the influence of more "local details" whereas a smaller k would conversely augment their weight throughout the alignment. Our experiments show that with partially overlapping point clouds a small list yields better results, while a list containing about 50% of total points is the better choice when both point clouds are the same.

We also presented a generalization of the weighting strategy (Eq. 12), that allows the use of multiple matching operators, each producing a multivariate dispersion of Euclidean points, represented by a cross-covariance matrix. The combination of these matrices represents the contributions of each dispersion.

In the proposed scoring system our method was within the best methods in all four point cloud tested, being competitive with the ICP-CTSF and the Sparse ICP with CTSF, both also using the dissimilarity factor between tensors to improve the matching.

As the initial angular displacements between both point clouds get wider, most

of the methods compared have their performance degraded. That does not happen with the SWC-ICP, as the method is invariant to the initial relative pose. Noise and outliers does not alter this behavior, despite the additional difficulty.

A drawback of our method is the necessity of a preprocessing stage to compute the tensors, that might be computationally expensive if the point cloud is too large, although parallelism techniques in this step are possible. The second drawback is the performance with small overlapping regions of the point clouds. The SWC-ICP requires over 50% of overlapping and less than 25% as individual region to show a reasonable result.

As future works, we believe that better algorithms to estimate the tensors in cases of partial overlapping may improve the scores of the SWC-ICP, as more correct correspondences yield better results. The CTSF does not consider relative orientation or curvature, for example. Additional voting fields can be used instead or after the isotropic field, to estimate tensors with such geometric features. Other matching operators might be included in our generalization using other invariant features, like second-order moments, curvature or spherical harmonics, as proposed by Sharp *et al.*[6]. Our evaluation protocol can be expanded with more challenging scenarios to the compared methods. Real range images and point clouds with different sampling rates are some examples found in the literature, but not yet unified in a quantitative evaluation protocol. Additional measurements might be interesting in specific applications, like computational time. We believe that advances towards a benchmark for rigid registration could be useful to upcoming researches.

Acknowledgments

The authors would like to thank CAPES and FAPEMIG for their financial support.

References

1. P. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell*, **14**(2), pp. 239–256 (1992).
2. S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *IEEE Proc. 3rd Int. Conf. 3-D Digital Imaging and Modeling*, 145–152 (2001).
3. J. Salvi, C. Matabosch, D. Fofi, and J. Forest, "A review of recent range image registration methods with accuracy evaluation," *Image Vis. Comput.*, **25**(5), 578–596 (2007).
4. Y. Díez, F. Roure, X. Lladó, and J. Salvi, "A qualitative review on 3d coarse registration methods," *ACM Computing Surveys (CSUR)*, **47**(3), 45 (2015).
5. G. C. Sharp, S. W. Lee, and D. K. Wehe, "Invariant features and the registration of rigid bodies," in *Proc. IEEE Int. Conf. Robotics and Automation, 1999*, IEEE Robotics and Automation Society, **2**, 932–937 (1999).
6. G. C. Sharp, S. W. Lee, and D. K. Wehe, "Icp registration using invariant features," *IEEE Trans. Pattern Anal. Mach. Intell*, **24**(1), 90–102 (2002).
7. A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Proc. Robotics: Science and Systems*, The MIT Press, **5** (2009).
8. J. Servos and S. L. Waslander, "Multi channel generalized-icp," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, 3644–3649 (2014).

9. L. Maier-Hein, A. M. Franz, T. R. dos Santos, M. Schmidt, M. Fangerau, H. Meinzer, and J. M. Fitzpatrick, "Convergent iterative closest-point algorithm to accomodate anisotropic and inhomogenous localization error," *IEEE Trans. Pattern Anal. Mach. Intell*, **34**(8), 1520–1532 (2012).
10. S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," *Comput. Graph. Forum*, **32**(5), 113–123 (2013).
11. P. Mavridis, A. Andreadis, and G. Papaioannou, "Efficient sparse icp," *Comput. Aided Geom. Des.*, **35**, 16–26 (2015).
12. J. Yang, H. Li, and Y. Jia, "Go-icp: Solving 3d registration efficiently and globally optimally," *IEEE Int. Conf. Computer Vision (ICCV)*. IEEE Comp. Soc., 1457–1464 (2013).
13. L. Reyes, G. Medioni, and E. Bayro, "Registration of 3d points using geometric algebra and tensor voting," *Int. Journal of Comput. Vis.*,**75**(3), 351–369 (2007).
14. B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell*, **33**(8), 633–1645 (2011).
15. L. W. X. Cejnog, "Rigid registration based on local geometric dissimilarity," Master's thesis, Post-Graduation Program on Computer Science, Federal University of Juiz de Fora, Juiz de Fora, MG, Brazil, 2015.
16. B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *JOSA A*, **4**(4), 629–642 (1987).
17. G. Medioni, C.-K. Tang, and M.-S. Lee, "Tensor voting: Theory and applications," *Proc. of Reconnaissance des Formes et Intelligence Artificielle, Paris, France*, **3** (2000).
18. P. Mordohai and G. Medioni, *Tensor Voting:A Perceptual Organization Approach to Computer Vision and Machine Learning.* Morgan & Claypool, **2**(1), 1–136 (2006).
19. M. Corsini, P. Cignoni, and R. Scopigno, "Efficient and flexible sampling with blue noise properties of triangular meshes," *IEEE Trans. Vis. Comput. Grphics*, **18**(6), 914–924 (2012).
20. M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans. Model. Comput. Simul. (TOMACS)*, **8**(1), 3–30 (1998).
21. N. Mellado, D. Aiger, and N. J. Mitra, "Super 4pcs fast global pointcloud registration via smart indexing," in *Comput. Graph. Forum*, **33**(5), 205–215 (2014).

Photo and Bibliography

Fernando Yamada has a graduate degree in Computer Science from Universidade Federal de Juiz de Fora (2014) and M.Sc. degree in Computer Science from Universidade Federal de Juiz de Fora in 2016.

Luciano Cejnog obtained his bachellor degree in Computer Science from Universidade Federal de Juiz de Fora in 2013, and M.Sc. degree in Computer Science from Universidade Federal de Juiz de Fora in 2015.

Marcelo Bernardes Vieira obtained a graduate degree in Computer Science from Pontifícia Universidade Católica de Minas Gerais (1995), M.Sc. in Computer Science from Universidade Federal de Minas Gerais (1998), Ph.D. in Image & Signal Processing from ENSEA/Université de Cergy-Pontoise in France (2002) and Ph.D. in Computer Science from Universidade Federal de Minas Gerais (2002), performed in conjunction. He was researcher at Instituto de Matemática Pura e Aplicada between 2003 and 2005. He is currently associate professor at Universidade Federal de Juiz de Fora. His main knowledge area is Computer Science, working on the following subjects: fluid dynamics, tensor fields and scientific visualization, physical systems geometric modeling, 3d reconstruction and pattern recognition. He is coordinator of the Group for Computer Graphics, Image and Vision at UFJF.

Rodrigo Silva obtained a graduate degree in Computer Science from Pontifícia Universidade Católica de Petrópolis (1999), M.Sc. in Systems Engineering and Computer Science from Universidade Federal do Rio de Janeiro (2002), Ph.D. in Civil Engineering from Universidade Federal do Rio de Janeiro (2006) and Postdoctorate in Computer Science from Laboratório Nacional de Computação Científica (2008). He is currently adjunct professor at Universidade Federal de Juiz de Fora and his main knowledge are is Computer Graphics with emphasis in Augmented Reality and Virtual Reality.