UNIVERSIDADE FEDERAL DE JUIZ DE FORA

INSTITUTO DE CIÊNCIAS EXATAS

PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Fernando Akio de Araujo Yamada

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

Juiz de Fora

2016

Fernando Akio de Araujo Yamada

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

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Rodrigo Luis de Souza da Silva

Coorientador: Marcelo Bernardes Vieira

Juiz de Fora

2016

Fernando Akio de Araujo Yamada

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

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Aprovada em 15 de Março de 2016.

BANCA EXAMINADORA

_____

Prof. D.Sc. Rodrigo Luis de Souza da Silva - Orientador
Universidade Federal de Juiz de Fora

_____

Prof. D.Sc. Marcelo Bernardes Vieira- Coorientador
Universidade Federal de Juiz de Fora

_____

Prof. D.Sc. Gilson Antônio Giraldi
Laboratório Nacional de Computação Científica

_____

Prof. D.Sc. Saul de Castro Leite
Universidade Federal de Juiz de Fora

ACKNOWLEDGMENTS

Fairy tales are more than true:
not because they tell us that
dragons exist, but because they
tell us that dragons can be
beaten. - G.K. Chesterson

RESUMO

No problema de registro rígido por pares é preciso encontrar uma transformação rígida que alinha duas nuvens de pontos. A solução clássica e mais comum é o algoritmo *Iterative Closest Point* (ICP). No entanto, o ICP e muitas de suas variantes requerem que as nuvens de pontos já estejam grosseiramente alinhadas. Este trabalho apresenta um método denominado *Shape-based Weighting Covariance Iterative Closest Point* (SWC-ICP), uma melhoria do ICP clássico. A abordagem proposta aumenta a possibilidade de alinhar corretamente duas nuvens de pontos, independente da pose inicial, mesmo quando existe apenas sobreposição parcial entre elas, ou na presença de ruído e *outliers*. Ela se beneficia da geometria local dos pontos, codificada em tensores de orientação de segunda ordem, para prover um segundo conjunto de correspondências para o ICP. A matriz de covariância cruzada computada a partir deste conjunto é combinada com a matriz de covariância cruzada usual, seguindo uma estratégia heurística. Para comparar o método proposto com algumas abordagens recentes, um protocolo de avaliação detalhado para registro rígido é apresentado. Os resultados mostram que o SWC-ICP está entre os melhores métodos comparados, com performance superior em situações de grande deslocamento angular, mesmo na presença de ruído e *outliers*.

Palavras-chave: Registro rígido. *Iterative Closest Point*. Tensor de orientação. Dissimilaridade de forma. Geometria computacional.

ABSTRACT

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 work a method named Shape-based Weighting Covariance Iterative Closest Point (SWC-ICP), an improvement over the classical ICP. Our approach improves the possibility to correctly align two point clouds, regardless of the initial pose, even when there is only a partial overlapping between them, 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 methods compared, with superior performance in situations of wide angular displacement, even in situations of noise and outliers.

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

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\vec{r}$	Vector $\vec{r} = (x, y, z)$, with $x, y, z \in \mathbb{R}$.
$\lVert \vec{r} \rVert$	L_2 Norm of a vector $\vec{r}$.
$\widehat{r}$	Normalized vector $\widehat{r} = (x, y, z)$, with $x, y, z \in \mathbb{R}$.
$\mathbf{S}_p$	Second-order orientation tensor associated with the point p.
$L_k(p)$	List of neighbors of a point p.
$L_k^{-1}(p)$	List of points that contains the point p as a neighbor.
M	Point cloud.
$p_i = (p_x, p_y, p_z)$	A point component of a point cloud.
$\mathring{q}_R$	A quaternion represented by a 4D vector $\mathring{q}_R = [q_0, q_1, q_2, q_3]^T$.

CONTENTS

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. All these problems use point clouds as data structures to store the spatial data. A point cloud is a set of points in the same coordinate system, in this case we are interested in points in $\mathbb{R}^3$. In the surface registration problem the goal is to align, or to register, two or more point clouds into a common reference system. In this work we focus on pairwise rigid registration, a sub-problem where a rigid transformation is sought to align two point clouds.

Usual problems include point clouds of the same scene only partially overlapped, point clouds corrupted by noise and outliers resulting from the sensor device, and incomplete models due to self-occlusion or scan range limitation. Since there is not a definitive solution to rigid registration in these scenarios 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). Our method uses the local geometric information of the points, coded in second-order orientation tensors, to improve the ICP transformation estimation step. The proposed approach increases the possibility to converge correctly for most cases, even when corrupted by additive noise and heavy amounts of outliers.

Several tests are performed following a proposed quantitative evaluation protocol, in order to give some statistical relevance to our results, when compared to some recent approaches. This protocol includes the description of the generation process of the simulated outliers, noise and partial overlapping point clouds. It also presents a scoring system to compare the results of each algorithm, in a more reliable manner than some protocols used in the rigid registration literature. We use a dataset composed by four point clouds of different topologies and geometries, with high and low curvature points, holes and some degree of symmetry.

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.

1.1   PROBLEM DEFINITION

Given two distinct point clouds named the Model set $M = \{m_i \mid m_i = (m_{ix}, m_{iy}, m_{iz})\}$, with $|M|$ points, and Data set $D = \{d_j \mid d_j = (d_{jx}, d_{jy}, d_{jz})\}$, with $|D|$ points, without any information about their relative position, the rigid registration problem is to find the rigid transformation $T(\mathbf{R}, \vec{t})$, $T : \mathbb{R}^3 \to \mathbb{R}^3$, where $\mathbf{R}$ is a rotation matrix and $\vec{t}$ is a translation vector, that applied to D best aligns both clouds, that is, a transformation that minimizes a distance metric, usually the Euclidean distance, between the homologous points in M and D.

The clouds M and D represent the same rigid object, or static scene, and both are supposed to have points sampled from at least one common surface of the object, i.e. a subset of them represents homologous regions in respect to the object. The clouds might have outliers and their points may be displaced by additive noise.

1.2   OBJECTIVES

The main objective is to improve the Iterative Closest Point (ICP) method, using the local geometric information of the points neighborhood to estimate a better transformation. Our goal is to increase the possibility to converge to the correct local optimum for an extended range of initial angles, in the presence of noise, outliers and different overlapping levels. Secondary objectives include:

- A quantitative evaluation protocol to rigid registration, able to compare different methods in a database composed by noise, outliers and partial overlapping point clouds, usual problems listed in the literature.

- The evaluation of the behavior of the SWC-ICP under different parameters using the proposed protocol and compare with other recent methods.

- The evaluation of how the different stages of the tensor estimation algorithm impact on the registration problem.

1.3 RELATED WORKS

The classical and most cited algorithm in the rigid registration literature is Besl and McKay's Iterative Closest Point (ICP) (BESL; MCKAY, 1992). This algorithm takes two point clouds, named Model and Data, and consists of the iteration of two major steps: matching and transformation estimation. In the matching step, the algorithm searches for every point in the Data set the closest point in the Model set. This set of correspondences is used to estimate a rigid transformation. These two steps are iterated until a termination criterion is satisfied. Almost at the same time, Yang and Medioni (1992) also developed a similar technique, taking advantage of the normal vectors at each point in order to build a more suitable corresponding set. This matching was named point-to-plane and has the advantage to converge with fewer iterations than the original.

Both approaches, although breakthrough at their time, presented several possible optimizations. They assume there is a correct correspondence for each point and the point clouds are already coarsely aligned. This assumption easily fails on real applications. Rusinkiewicz and Levoy (2001) proposed in their survey a classification of six possible stages of the algorithm that could be optimized, and most works improving the ICP can still be fit in their classification. Another survey, written by Salvi et al. (2007), covers the rigid registration area splitting the methods in coarse and fine registration, giving the strengths and weaknesses of each.

Coarse registration includes methods that aim to give a good initial guess of the rigid transformation between two sets. These methods usually try to be more robust to noise, outliers and large initial displacement, at the cost of a precise result. Some of these methods can also be applied to real time applications due to low computational cost. Díez et al. (2015) further classify coarse methods according to a proposed pipeline, composed by three stages of optimizations: keypoints detection, description and search strategy. Methods improving the first stage (DONOSER; BISCHOF, 2006; SUN et al., 2009; ZAHARESCU et al., 2009; ZHONG, 2009; MIAN et al., 2010) aim to select a subset of points in one or both point clouds, in order to speed up the registration process or provide significant points to next steps. Description methods (JOHNSON; HEBERT, 1999; POTTMANN et al., 2009; TOMBARI et al., 2010; OVSJANIKOV et al., 2010) try to assign local descriptors to the points or build a histogram, according to some feature estimated. Finally, methods of the third stage (CHEN et al., 1999; GELFAND et al.,

2005; AIGER et al., 2008; MELLADO et al., 2014; ALBARELLI et al., 2015) employ a refined strategy attempting to reduce the search space.

Fine methods on the other hand, assume the point clouds are already pre-aligned and aim to find the most accurate result as possible. Most of fine methods are based on modifications of the ICP algorithm. Godin et al. (1994) use the intensity information, provided by some range scanners, to improve the number of correct matches, naming their method Iterative Closest Compatible Point. Sharp et al. (1999, 2002) 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 MSE on each iteration. Some authors attach local covariance matrices to the points, in a way that the local structure is considered. The Generalized-ICP (SEGAL et al., 2009) and Multi-Channel Generalized-ICP (SERVOS; WASLANDER, 2014) are some examples, with the latter using additional channels other than its position to improve the matching step and the covariance estimation. The A-ICP (MAIER-HEIN et al., 2012) accounts the anisotropic localization error into a covariance matrix, representing a local zero-mean Gaussian distribution, that is used in both the matching step and transformation estimation. 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 to find the optimal transformation. Fitzgibbon (2003) used the Levenberg-Marquardt in his LM-ICP, a well-known general purpose non-linear optimization method, combined with the Distance Transform data structure, which provide fast access to the nearest neighbors search and eases the calculation of derivatives needed for the Levenberg-Marquardt. As noted by the author, these non-linear strategies may demand more computational time than specialized algorithms, such as the ICP, but with the speedups used, the LM-ICP was still comparable to other methods. Bouaziz et al. (2013) 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 resilient to noise and outliers, however the time required is unfeasible. Mavridis et al. (2015) relieves this problem adapting a Simulated Annealing process before the ADMM optimization, which gives a good initial guess to the ADMM. Yang et al. (2013) 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. (2007) presented a different approach, not based on the ICP, to solve rigid registration using geometric algebra. 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. A novel approach for surface registration, other than ICP-based, is the game-theoretical framework proposed by Albarelli et al. (2015). In this work, a set consisted of candidate matches are selected and compete to each other until the game reaches an equilibrium. The remaining points are then used to estimate the transformation.

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, and the absence of benchmarks or standard datasets also difficult the comparison between methods. Noise, outliers, occlusion and partial overlapped meshes are usual problems addressed. While these are valid scenarios to show their potential improvements, these papers lack some statistical analysis. Time, residual error and number of iterations are the common metrics, but their relevance cannot be guaranteed with just few examples that might be biased by the initial applied transformation. We understand that quantitative analyses are computationally expensive for rigid registration, especially with huge point clouds. Too much computational time is required to run several trials, with different parameters and compare with other algorithms, even with the current parallelism technologies.

Although this kind of analysis is rare, we highlight some examples found in the literature. Sharp et al. (1999) ran 100 random transformations and presented a table with the convergence rate to a location near the ground truth defined. Reyes et al. (2007) 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 and Vemuri (2011) measured the convergence range angles for 2D rigid registrations, and also success rates for 3D partial range images, with 30 different transformations. The random nature of the transformations used reflects better the

performance of the method in real scenarios.

Most of this work was developed together with Cejnog's dissertation (CEJNOG, 2015), and we share many ideas, like the Comparative Tensor Shape Factor (CTSF). In Cejnog's work, the CTSF is used in the ICP matching step, with a linear combination very similar to what was used in Sharp et al. (2002), but the weight α is updated only when a local minimum is reached, instead of every iteration. Here we use the same base strategy, but our proposal is to modify the transformation estimation instead, so that it also considers the local geometry of the points. The tensors we use are estimated using a framework similar to Tensor Voting (MEDIONI et al., 2000), based in the work of Vieira et al. (2004), that uses two different voting fields. The first is based only on the Euclidean distance of the points, and the second uses a coplanarity constraint to reinforce planar structures. However, unlike Cejnog and Vieira, we use only the first voting field. As results show, this step alone produces better results on the matching step of the ICP. The second-order tensors used consist of a covariance matrix associated with a locality in $\mathbb{R}^3$, and encodes the local geometry. The SWC-ICP, like Cejnog's ICP-CTSF, is a coarse-to-fine method, able to roughly align two point clouds with wide angular displacement, and its behavior smoothly changes into a fine method, hence, able to reach fine levels of registration. The presented experiments include noise, outliers and partial overlapping cases, showing that the SWC-ICP have a broader range of convergence than the ICP-CTSF and other recent approaches.

2 FUNDAMENTALS

2.1 ORIENTATION TENSOR FOR GEOMETRY ESTIMATION

Tensors are a generalization of the concept of scalar and vector. They have been applied in many areas such as engineering, physics and computer vision. We use the orientation tensor, a second-order tensor, represented as a $n \times n$ covariance matrix, associated with a locality in $\mathbb{R}^n$. In our context we use $n = 3$, since we are dealing with 3D points. Because the orientation tensor is a covariance matrix, it is symmetric, positive semidefinite and its eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$ are greater or equal to zero. The orientation tensor can be decomposed in three parts, regarding its linear, planar and spherical components contribution:

$$\mathbf{T} = (\lambda_1 - \lambda_2)\mathbf{T}_1 + (\lambda_2 - \lambda_3)\mathbf{T}_2 + (\lambda_3)\mathbf{T}_3.$$

The tensor $\mathbf{T}_1$ corresponds to the linear component, and its main direction coincides with the direction of the eigenvector $\vec{e_1}$ associated to the main eigenvalue λ_1 of $\mathbf{T}$. $\mathbf{T}_2$ corresponds to the planar component, formed by the two main directions $\vec{e_1}$ and $\vec{e_2}$. Finally, $\mathbf{T}_3$ is the spherical component, formed by all three main directions.

The anisotropy coefficients of the orientation tensor c_l, c_p and c_s (WESTIN et al., 1997) are calculated using the eigenvalues of the covariance matrix. Since the eigenvalues λ_1, λ_2 and λ_3 of the matrix are sorted and normalized, i.e., $\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2} = 1$, three cases are possible: one eigenvalue is much higher than the other two, one eigenvalue is much smaller than the other two, and the three eigenvalues are equal, each case yielding a high value of c_l, c_p and c_s, respectively. Their equations are:

$$c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3}, \quad c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3}, \quad c_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}.$$

Figure 2.1 shows superquadric glyphs (KINDLMANN, 2004) varying the proportion between its eigenvalues, reflecting in different shapes and anisotropy coefficients. The figures on the rest of this work that represent graphically second-order tensors follow that glyph representation.

Vectors can be converted to orientation tensors following the product $\mathbf{T}_v = \vec{v} \cdot \vec{v}^T$,
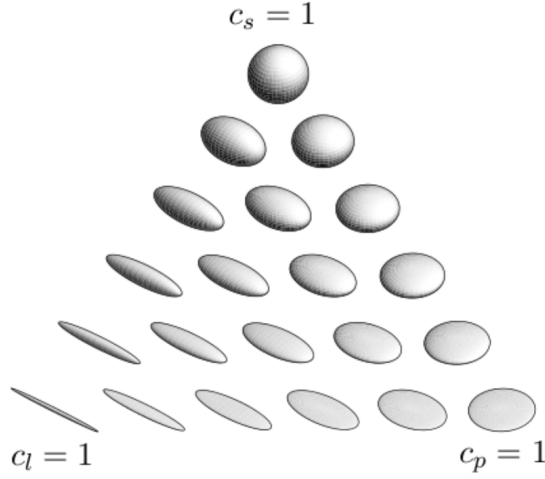
Figure 2.1: Ellipsoid tensors with different predominant anisotropy coefficients. A tensor with high c_l is called a stick tensor, high c_p indicates a plate tensor, and high c_s a ball tensor. Image from Kindlmann (2004).

considering $\vec{v}$ a column vector. The tensors built in this way have two zero eigenvalues, and the main eigenvector has the same direction of $\vec{v}$. It has a large linear coefficient c_l, thus yielding a stick shape.

Opposing vectors when summed cancel themselves, but the sum of the tensors of these vectors do not. Thus, tensors are more suitable than vectors to accumulate geometric information. Regarding the directions, consider, for example, two orthogonal vectors $\vec{a}$ and $\vec{b}$. The sum of their tensors $\mathbf{T}_a$ and $\mathbf{T}_b$ produces a tensor with large planar coefficient c_p, and the two main eigenvectors λ_1 and λ_2 have the directions of $\vec{a}$ and $\vec{b}$. Conversely, if $\vec{a}$ and $\vec{b}$ are collinear, the resulting tensor of their sum has the shape of a stick, and the same direction of $\vec{a}$ and $\vec{b}$. Figure 2.2 exemplifies this sum.

The method to estimate the local geometry, also presented by Cejnog (CEJNOG, 2015), is based on the method proposed by Vieira et al. (2004) and is very similar to the Tensor Voting framework (MEDIONI et al., 2000; MORDOHAI; MEDIONI, 2006). It encodes in tensors the geometric disposition of the neighborhood of each point. Each point casts its influence on the neighborhood through a vector voting field. Each vote is converted to a tensor and accumulated on the neighbors. The resulting tensors represent the local geometry, and once estimated, it does not change under rigid transformations, making it an invariant feature suitable for rigid registration.

Since the method only deals with the position of the points, higher order elements like orientation and curvature sign are not identified. Tensors at the peak of concavities and

$$\vec{a} = (0\ 1\ 0)^T \qquad \mathbf{T}_a = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \times (0\ \ 1\ \ 0)$$

$$\mathbf{T}_a = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$\lambda_1 = 1 \quad \vec{e_1} = (0\ 1\ 0)$
$\lambda_2 = 0$
$\lambda_3 = 0$

$$\vec{b} = (1\ 0\ 0)^T \qquad \mathbf{T}_b = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \times (1\ \ 0\ \ 0)$$

$$\mathbf{T}_b = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$\lambda_1 = 1 \quad \vec{e_1} = (1\ 0\ 0)$
$\lambda_2 = 0$
$\lambda_3 = 0$

$$\mathbf{T}_a + \mathbf{T}_b = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$\lambda_1 \approx 0.7 \quad \vec{e_1} = (1\ 0\ 0)$
$\lambda_2 \approx 0.7 \quad \vec{e_2} = (0\ 1\ 0)$
$\lambda_3 = 0$
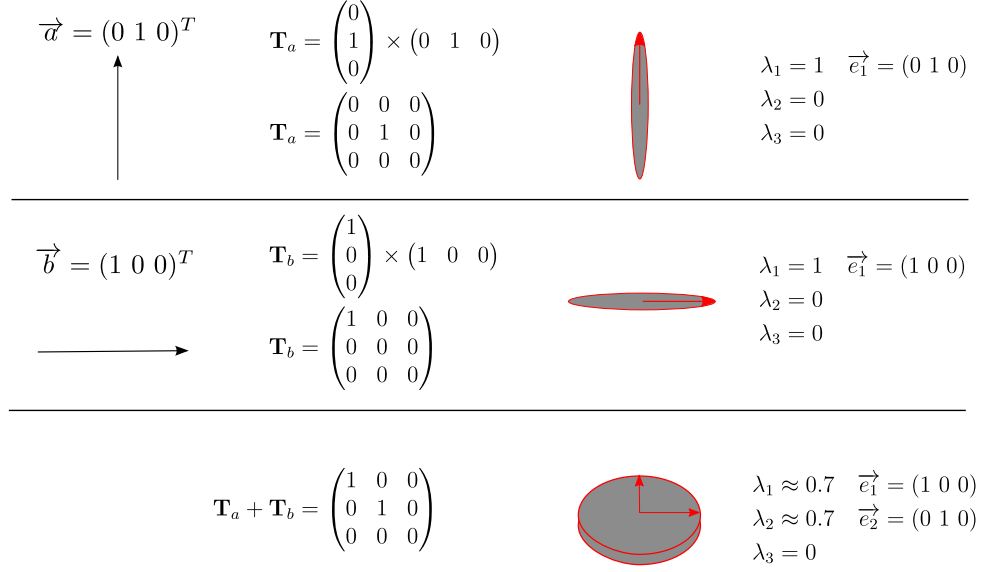
Figure 2.2: Two vectors $\vec{a}$ and $\vec{b}$, with its respective tensors $\mathbf{T}_a$ and $\mathbf{T}_b$, and the tensor resulting from the sum.

convexities might be equal if estimated using few neighbors. But with a sufficiently large number of neighbors, the estimation tends to be global, and except for a high degree of symmetry, these tensors will be different. However, in the context of the rigid registration problem, these cases are uncommon in real scenarios.

Cejnog (2015) uses 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 then must be applied after the first. It enhances the approximation of the geometry enforcing coplanar structures. A different distance metric is used in the second, based on the distance over an elliptical trajectory. As a consequence of the second field, the shape of the tensors tends to be more planar, raising the mean c_p of the point cloud. But differently from 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 information about triangles, the neighborhood of a point p is represented by a list $L_k(p)$ of its k nearest-neighbors sorted by their Euclidean distances. We denote $L_k^{-1}(p)$ as the set of points containing p in their list. 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.

2.1.1   ISOTROPIC VOTING FIELD

The isotropic voting field (CEJNOG, 2015) builds for each point p a second-order tensor $\mathbf{T}_p$, which accumulates the weighted sum of tensors built from the vote vectors $\overrightarrow{qp}$, $\forall q \in L_k(p)$.

The stick tensors $\mathbf{T}_{qp}$ formed by the product of the normalized $\widehat{qp} \cdot \widehat{qp}^T$, 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.

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

The output is the tensor $\mathbf{T}_p$:

$$\mathbf{T}_p = \sum_{q \in L_k(p)} e^{\frac{-||\overrightarrow{pq}||^2}{\sigma_p^2}} \cdot \widehat{pq} \cdot \widehat{pq}^T. \tag{2.1}$$

Equation 2.1 is analogous to the application of a 3D isotropic radial structuring element to the point cloud.

The eigenvector $\overrightarrow{e_3}$, associated to the third main eigenvalue λ_3 of $\mathbf{T}_p$, is an approximation of the normal at p, being more accurate in points with planar neighborhood.

Figure 2.3 and 2.4 show the tensors estimated with different values of k. As the number of neighbors increase, the main direction of the tensors, especially at the top and the bottom, are pointed to the center of the point cloud. The presence of noise affects directly the tensors, especially with low values of k. As we can see in Figure 2.4 (c), the tensors obtained with $k = 1\%$ are very different of the ones obtained with the clean point cloud of Figure 2.3 (a). The extra anisotropic voting field used by Cejnog (2015) is presented in the Appendix A.

(a) k=1%.　　(b) k=5%.　　(c) k=10%.　　(d) k=50%.　　(e) k=75%.　　(f) k=100%.
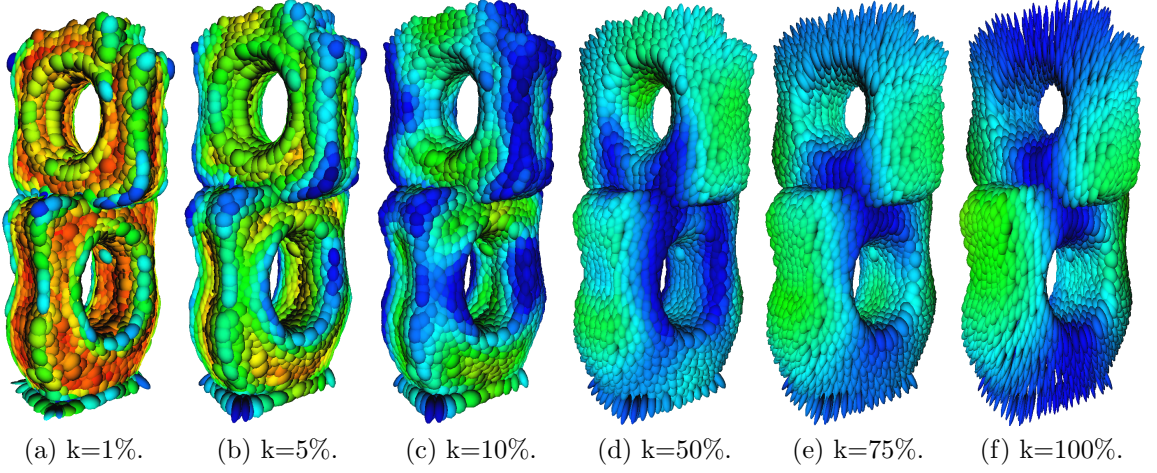
Figure 2.3: Variation of the nearest neighbors list size using the first voting field. (a): 1% of the total points. (28 neighbors) (b): 5% of the total points. (136 neighbors). (c): 10% of the total points. (272 neighbors). (d): 50% of the total points. (1356 neighbors). (e): 75% of the total points. (2034 neighbors). (f): 100% of the total points. (2711 neighbors). The colors represent the c_p of the points: blue tensors have stick or ball shapes while red tensors have plate shapes.


(a) k=1%.　　　　(b) k=100%.　　　　(c) k=1%.　　(d) k=100%.
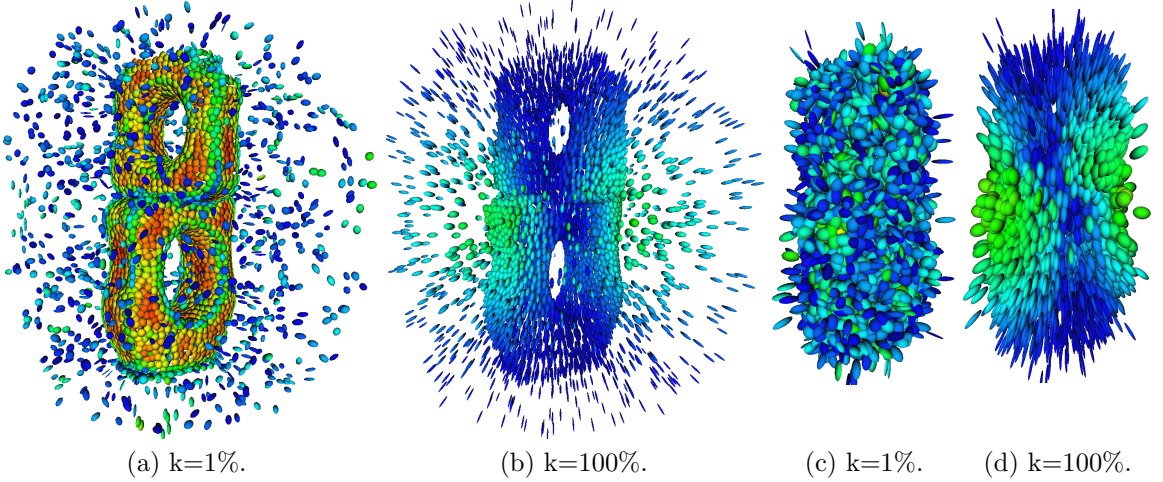
Figure 2.4: Variation of the nearest neighbors list size using the first voting field, with outliers and noise. (a): 1% of the total points, with 50% of outliers. (b): 100% of the total points, with 50% of outliers. (c): 1% of the total points, with 5% of noise. (d): 100% of the total points, with 5% of noise. The colors represent the c_p of the points: blue tensors have stick or ball shapes while red tensors have plate shapes.

2.2　COMPARATIVE TENSOR SHAPE FACTOR

The Comparative Tensor Shape Factor (shortened CTSF)(CEJNOG, 2015) is used whenever the shape of two tensors must be compared. Considering the representation of a tensor by a hyper-ellipsoid, its axes are proportional to the eigenvalues of its matrix, with

the larger eigenvalue associated to the largest axis, the second eigenvalue to the second larger axis, and so on. In this way, to compare the eigenvalues between two tensors is equivalent to compare the shapes of the respective tensors. Since what matters is the shape of the hyper-ellipsoid, 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:

$$CTSF(\mathbf{S}_1, \mathbf{S}_2) = \sum_{i=1}^{N} \left(\lambda_i^{\widehat{\mathbf{S}}_1} - \lambda_i^{\widehat{\mathbf{S}}_2} \right)^2,$$

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.
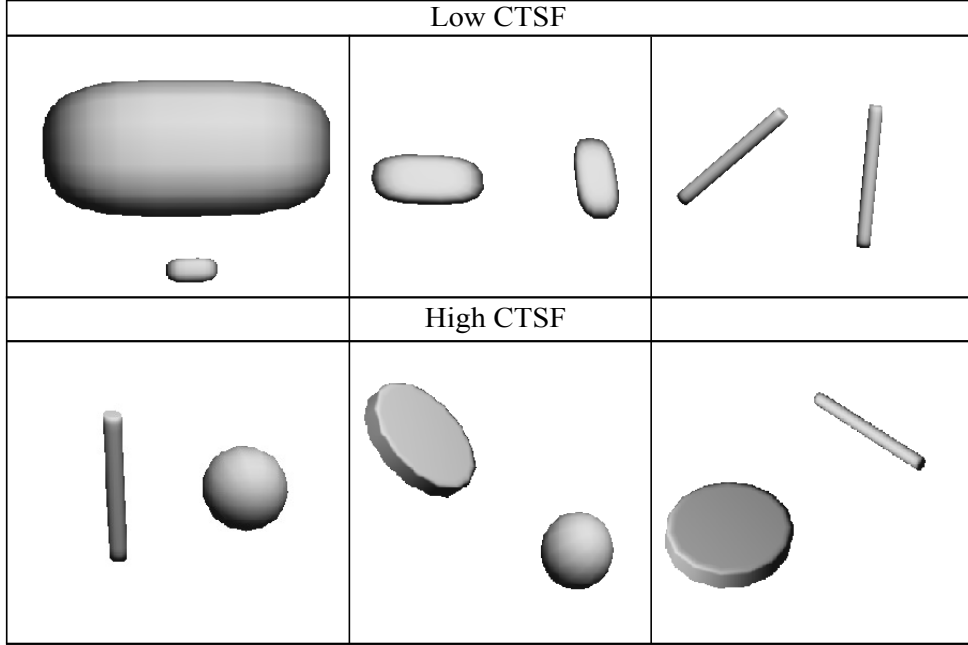

Figure 2.5: Examples of the CTSF Between Two Tensors. Figure from Cejnog (2015).

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 in a coarse matching scheme. Figure 2.5 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.

2.3 ITERATIVE CLOSEST POINT

Besl and McKay's ICP (BESL; MCKAY, 1992) 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, and are iterated until a stopping criterion is satisfied. 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 closest point operator, with respect to the Euclidean distance, 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, which is 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:

$$\min_{\mathbf{R}, \vec{\mathbf{t}}} \left(\sum_{i=1}^{|D|} ||e_i - \mathbf{R} \cdot d_i - \vec{\mathbf{t}}||^2 \right). \tag{2.2}$$

The original ICP estimates transformations in $\mathbb{R}^3$ using the method of Unit Quaternions, a closed-form solution developed by Horn (1987). Quaternions are a convenient mathematical tool to express rotations in 3D and are represented by a 4D vector $\mathring{q}_R = [q_0, q_1, q_2, q_3]^T$. For a unit quaternion it follows that $q_0 > 0$ and $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. q_0 is said the scalar part, and q_1, q_2, q_3 are the imaginary vector part. The scalar part is associated with the magnitude of the rotation, while the vector part is the rotation axis.

So in order to find the rotation that minimizes Equation 2.2, the method needs to find a quaternion instead. First, the centroids of each set are calculated:

$$\mu_D = \frac{1}{|D|} \sum_{i=1}^{|D|} d_i, \quad \mu_E = \frac{1}{|E|} \sum_{i=1}^{|E|} e_i. \tag{2.3}$$

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 is given by:

$$\Sigma_{DE} = \frac{1}{|D|} \sum_{i=1}^{|D|} [d_i e_i^T] - \mu_D \mu_E^T, \tag{2.4}$$

and the matrix L is defined as:

$$L = \begin{bmatrix} \Sigma_{11} + \Sigma_{22} + \Sigma_{33} & \Sigma_{32} - \Sigma_{23} & \Sigma_{13} - \Sigma_{31} & \Sigma_{21} - \Sigma_{12} \\ \Sigma_{32} - \Sigma_{23} & \Sigma_{11} - \Sigma_{22} - \Sigma_{33} & \Sigma_{12} + \Sigma_{21} & \Sigma_{31} + \Sigma_{13} \\ \Sigma_{13} - \Sigma_{31} & \Sigma_{12} + \Sigma_{21} & -\Sigma_{11} + \Sigma_{22} - \Sigma_{33} & \Sigma_{23} + \Sigma_{32} \\ \Sigma_{21} - \Sigma_{12} & \Sigma_{31} + \Sigma_{13} & \Sigma_{23} + \Sigma_{32} & -\Sigma_{11} - \Sigma_{22} + \Sigma_{33} \end{bmatrix}. \tag{2.5}$$

The eigenvector associated with 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 Equation 2.2. The proofs and expanded equations of this method can be found in Horn's paper (HORN, 1987). Comparisons and other details are presented by Eggert et al. (1997). 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:

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

With $\mathbf{R}$ and $\vec{\mathbf{t}}$ computed, the transformation is applied to all points of 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.

| (a) Initial pose. | (b) After 2 iterations. | (c) After 4 iterations. |

| (d) After 6 iterations. | (e) After 9 iterations. | (f) Final alignment after 11 iterations. |
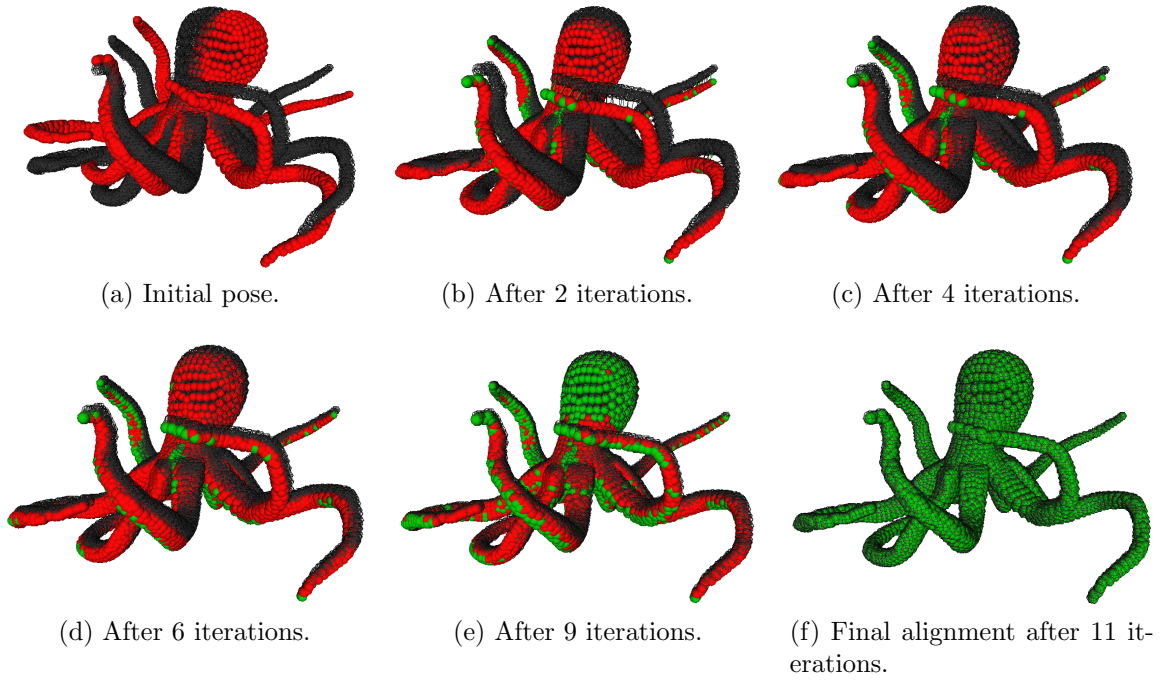
Figure 2.6: A correct convergence sequence of the ICP for an initial pose rotated 15°. Red points represent the Data and Black represent the Model. Green points indicate a point in Data matched correctly.

Besl and McKay assume that initially both Model and Data point clouds are coarsely aligned, i.e., the angular displacement between them is already low, whether by the result of an algorithm or manually positioning them. Figure 2.6 shows a convergence sequence of this case. When coarsely aligned many correspondences should already be correct, as observed in the tentacle of the octopus in Figure 2.6 (b), and at the first iterations

(a) Initial pose.   (b) After 4 iterations.   (c) After 8 iterations.

(d) After 13 iterations.   (e) After 18 iterations.   (f) Final alignment after 21 iterations.
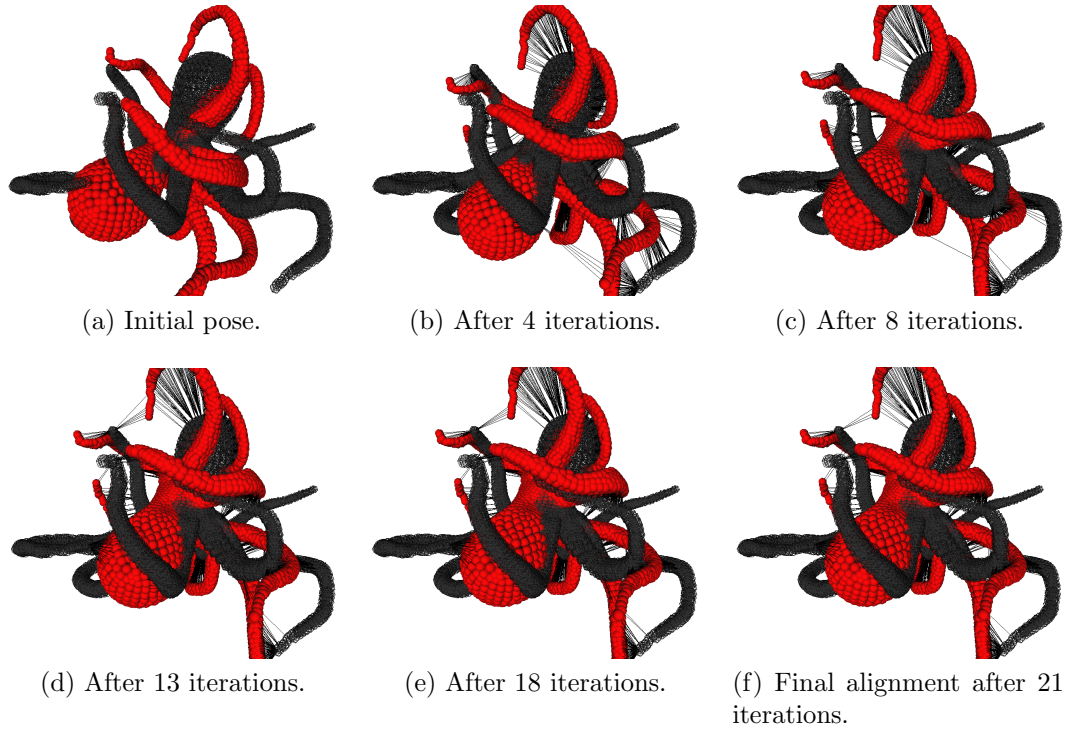
Figure 2.7: A failed convergence sequence of the ICP for an initial pose rotated 180°. Note that across the iterations one point in the Model set still has many correspondences on the Data set. Red points represent the Data and Black represent the Model.

the method will approximate these correspondences to the Data set. 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. Once the matching assigns too many false correspondences, see Figure 2.7, the estimation will lead to a relative pose even worse than the previous. This vicious cycle ends when a local optimum is reached, but too far from the global one.

It is clear now that better matchings or strategies that guide the transformation towards a better solution improve the chances to converge to the desired optimum. Indeed, most existent approaches attack these two problems. Coarse registration methods, for example, focus on providing a better initial pose to the ICP. Our approach besides providing a better initial pose, is also able to finely register the point clouds without the need of an additional algorithm.

2.4   SHAPE-BASED MATCHING

Following the premise that a better matching leads to a better transformation, the ICP-CTSF (CEJNOG, 2015) implements a matching strategy using an invariant feature to improve the number of correct correspondences. The feature chosen is the shape of the tensor at each point.

The CTSF was presented in Section 2.2 as a comparison factor between two tensors and 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 is the same. In this way, the CTSF between them should be very low. So no matter how apart a point cloud is from the other, the shape-based matching will be able to produce a good correspondences set, since only the shapes of the tensors are considered. On the other hand, if the point clouds are far from each other, the Euclidean match would result in a bad correspondences set, with a few points being matched to many. This behavior is observed very often in the first iteration of the classical ICP. Even though this is not a problem in some cases, it still has a great chance to converge to a local optimum far from the best depending on the initial relative pose.

However, 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 optimum very fast, often just coarsely aligned. Naturally the Euclidean matching is not static, improving its quality as long as the point clouds are getting aligned. As already known by the literature, if the point clouds are very close to each other the ICP is able to finely register them.

Two distinct patterns are observed in these two strategies. The shape-based matching provides a good coarse alignment no matter the initial relative pose, and the Euclidean match is better when point clouds are close. So the ICP-CTSF combine them taking the best of each one. Since the goal is to be able to align point clouds with large angular displacement, the shape-based matching should have more relevance in the first iterations. This is the coarse phase of the method. Once it reaches a local optimum hopefully better than the initial pose, it should now have a better Euclidean matching, making it deserve a greater influence. Throughout the iterations the influence of the Euclidean distance increases, becoming more similar to a fine method. 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 the number of local optima reached. The parameter w_0 is the initial weight given to the CTSF and a high value is the usual choice. b controls the update size of the weighting factor and impacts directly in the results of the ICP-CTSF. It is, however, a trade-off between robustness and time. A value too small will give more importance to Euclidean distance too fast and the fine stage of the method might not have a good relative pose to start. A value too high will guarantee that the point clouds are coarsely aligned, but implies in a large amount of iterations, since the update rate of the weight is too small. To avoid numerical instabilities we set $w_n = 0$, when $w_n \approx 0$.

3 PROPOSED METHOD

The proposed method also follows the same coarse-to-fine idea of the ICP-CTSF. Points with similar shapes should have more importance in the beginning of the registration, and the Euclidean distance should have more importance in the end. The difference from the previous method (CEJNOG, 2015) is the location of the selected correspondent points.

Our method starts approximating the point clouds reducing the distance of points whose tensor shape are similar, according to the CTSF, giving little relevance to the Euclidean distance between them. Again, what matters in the beginning is the coarse alignment. Then, following the same strategy of the shape-based matching, when a local optimum is reached, the weighting factor is recalculated and the CTSF gets less influence. After a sufficiently large number of local optima, the behavior is the same as a fine method, since the influence of the CTSF becomes insignificant.

We apply this strategy directly over the covariance matrix between the Data set and the correspondences set, using two covariance matrices, one established by the Euclidean distance of the closest points and one by the distance of points with low CTSF.

The cross-covariance of the points obtained with the Euclidean distance is computed exactly in the same way of the original ICP. The only difference of the second cross-covariance is the correspondences set used, 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 shape is the most similar to d_i, according to the CTSF of both. Like in Equation 2.3, the centroid of this shape-based correspondences set is:

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

If we minimize only the distance between points whose shapes are equals, the cross-covariance matrix would be expressed by:

$$\Sigma_{DS} = \frac{1}{|D|} \sum_{i=1}^{|D|} [d_i s_i^T] - \mu_D \mu_S^T. \tag{3.1}$$

Although this registration alone would not be correct according to the problem statement, it does contribute to reach a better alignment when combined to the original one.

Therefore, our contribution is a sum of both Equations 2.4 and 3.1, weighted by a factor w_n:

$$\Sigma_{DES} = \frac{1}{|D|} \sum_{i=1}^{|D|} [d_i e_i^T + w_n d_i s_i^T] - [\mu_D \mu_E^T + w_n \mu_D \mu_S^T]$$

$$= \frac{1}{|D|} \sum_{i=1}^{|D|} [d_i(e_i^T + w_n s_i^T)] - [\mu_D(\mu_E^T + w_n \mu_S^T)],$$

$$w_n = w_0 b^n, \ b < 1 \text{ and } 0 \leq w_n < w_0.$$

Similar to the heuristic presented in Section 2.4, n is incremented when a local optimum 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.

The matrix L of Equation 2.5 is computed as usual, with Σ_{DES} instead of Σ_{DE}, from Equation 2.4. The rotation and the translation are also computed exactly in the same way of the original ICP. The pseudocode of the SWC-ICP is in the Algorithm 1, of the Appendix B.

Analyzing the proposed formulation, we note that instead of using the closest point or the point whose shape of the tensor is the most similar, we use a combination of both. This means that the point considered in the correspondences set starts in a fictitious location and ends being the closest point. Results show that this approach has a higher rate of convergence to the correct optimum than some recent methods. Figure 3.1 shows the convergence sequence of the SWC-ICP for the same initial pose of Figure 2.7.

Unlike the method proposed in Cejnog (2015), in this work we do not use the anisotropic voting field to estimate the tensors. We conducted experiments of how each voting field impacts the registration in the final stages of this work, in collaboration with Luciano Cejnog. We observed that the results improved up to 30% using only the isotropic field. Therefore, we chose not to include the second voting field into our method.

Although the second voting field gives a better estimation of local planar structures, we are interested in the distinctiveness of each tensor during the matching step of the ICP. Figure 3.2 shows a histogram of the value obtained by the CTSF between all points of the Data point cloud and all other Model points during the matching step, considering a case with total overlapping and corrupted with noise and outliers.

We see values of CTSF more distributed when only the first voting field is used. Also, less points have lower values of CTSF, reducing the possibility of wrong correspondences.

(a) Initial pose.　(b) After 1 iteration.　(c) After 6 iterations.

(d) After 9 iterations.　(e) After 13 iterations.　(f) Final alignment after 17 iterations.
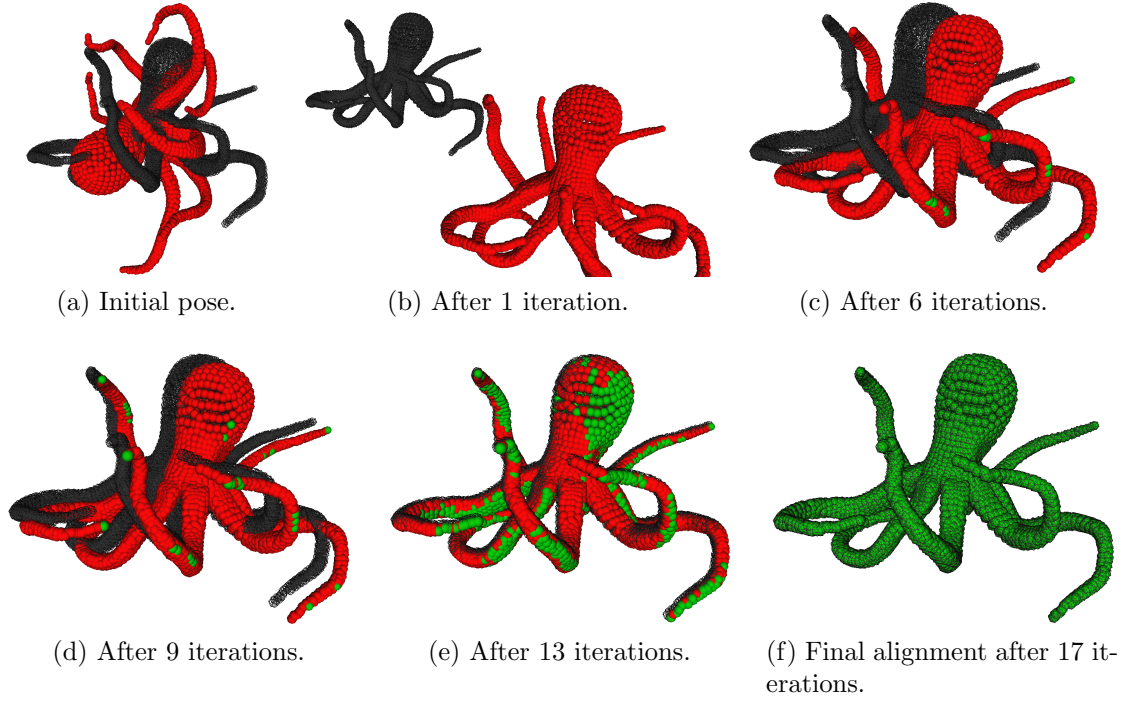
Figure 3.1: A convergence sequence of the SWC-ICP for an initial pose rotated 180°. The first iteration of the SWC-ICP moves away the centroid of the Data point cloud, but already corrects its orientation. Red points represent the Data and Black represent the Model. Green points indicate a point in Data matched correctly.


CTSF distribution using both voting fields

(a) $k = 1\%$.　(b) $k = 5\%$.　(c) $k = 15\%$.　(d) $k = 50\%$.　(e) $k = 75\%$.　(f) $k = 100\%$.

CTSF distribution using only the first voting field

(a) $k = 1\%$.　(b) $k = 5\%$.　(c) $k = 15\%$.　(d) $k = 50\%$.　(e) $k = 75\%$.　(f) $k = 100\%$.
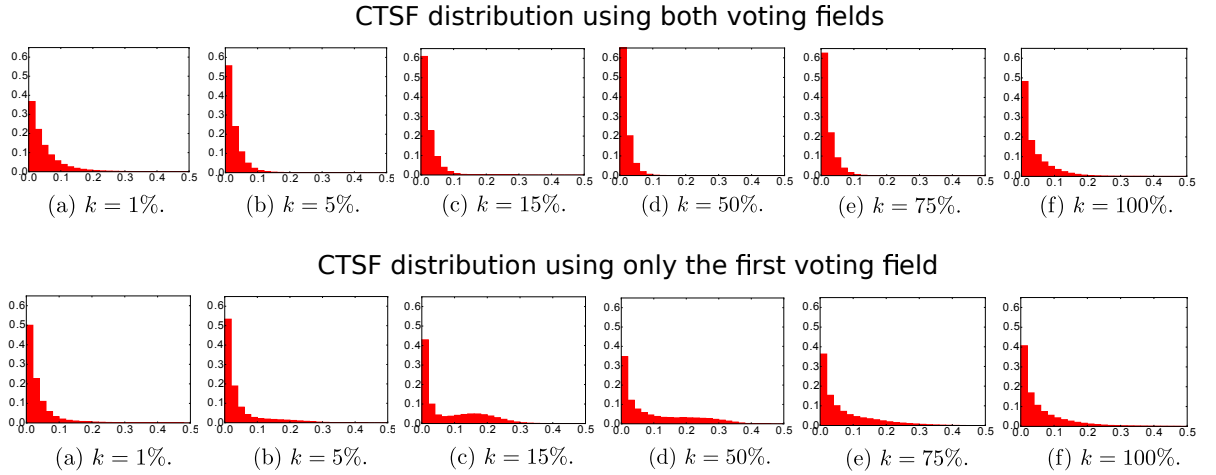
Figure 3.2: The histograms of CTSF between two point clouds corrupted with noise and outliers, with different sizes of k-neighborhood. Note that as the size increases, the number of values on the first bin is smaller when just the first voting field is used. Therefore, the likelihood of wrong correspondences in the matching step of the ICP is reduced.

Since the second field constrains some neighbors to cast its vote, and it reinforces the planarity of the tensor, the variety of shapes decrease. This is particularly seen in points over smooth surfaces. While the first step alone might not be enough to approximate the local curvatures of the point cloud, it allows a greater diversity of shapes.

4 EXPERIMENTAL RESULTS

4.1 3D MODEL DATASET SETUP

(a) Bunny.   (b) Happy Buddha.   (c) Octopus.   (d) Genus-2.
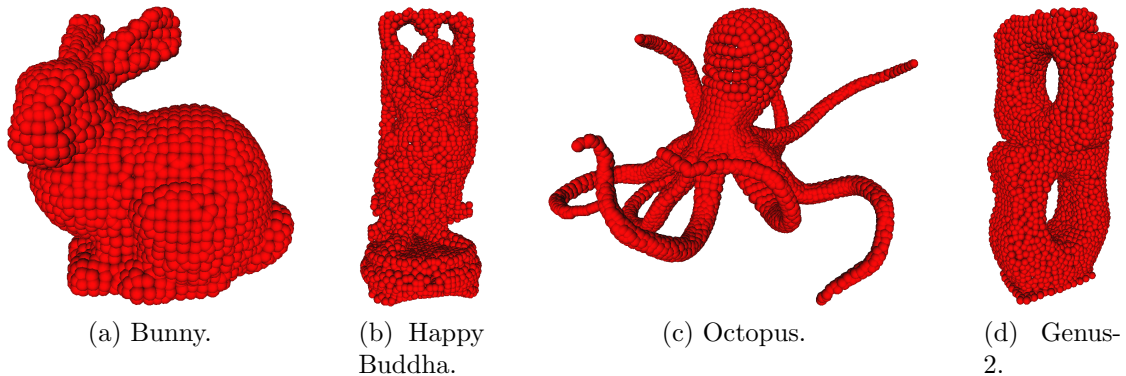
Figure 4.1: The point clouds used on the experiment: (a): Bunny (containing 1889 points). (b): Happy Buddha (3118 points). (c): Octopus (3822 points). (d): Genus-2 (2711 points).

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 point clouds: Bunny[1], Happy Buddha[1], Octopus[2] and Genus-2[3], with 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

[1]Provided by Stanford University Computer Graphics Laboratory on `http://graphics.stanford.edu/data/3Dscanrep/` .

[2]Provided courtesy of INRIA by the AIM@SHAPE-VISIONAIR Shape Repository.

[3]Provided by École Polytechnique Fédérale de Lausanne Computer Graphics and Geometry Laboratory on `http://lgg.epfl.ch/statues_dataset.php`

other point clouds were sampled using a Poisson-disk sampling algorithm (CORSINI et al., 2012) available on MeshLab[4]. All point clouds used are normalized with the largest edge of the bounding box scaled to have size 1. Figure 4.1 shows the point clouds used.

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 with 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 is 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. 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 by a scale δ that controls the intensity of the noise. We use an isotropic displacement vector because it is more challenging than noise in the direction of the normal:

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

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 overlapped region.

The values for noise used are: $\delta = \{0.01, 0.05\}$. Since the point clouds are normalized, the noise is correspondent to 1% and 5% of the edge of a unit box. Higher amounts of noise are not considered because the point cloud loses its features and such level of degradation is an unrealistic case. Outliers are tested with 5%, 20%, 50% of the 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. Figure 4.2 shows some examples of point clouds corrupted with outliers, noise and both at the same time. Note that although 1% of noise does not alter much of the geometry, with 5% the point cloud already loses most of its details.

---

[4] http://meshlab.sourceforge.net

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

(a) 10% of outliers. (b) 20% of outliers. (c) 50% of outliers.

(d) 1% of noise. (e) 5% of noise. (f) 50% of outliers and 5% of noise.
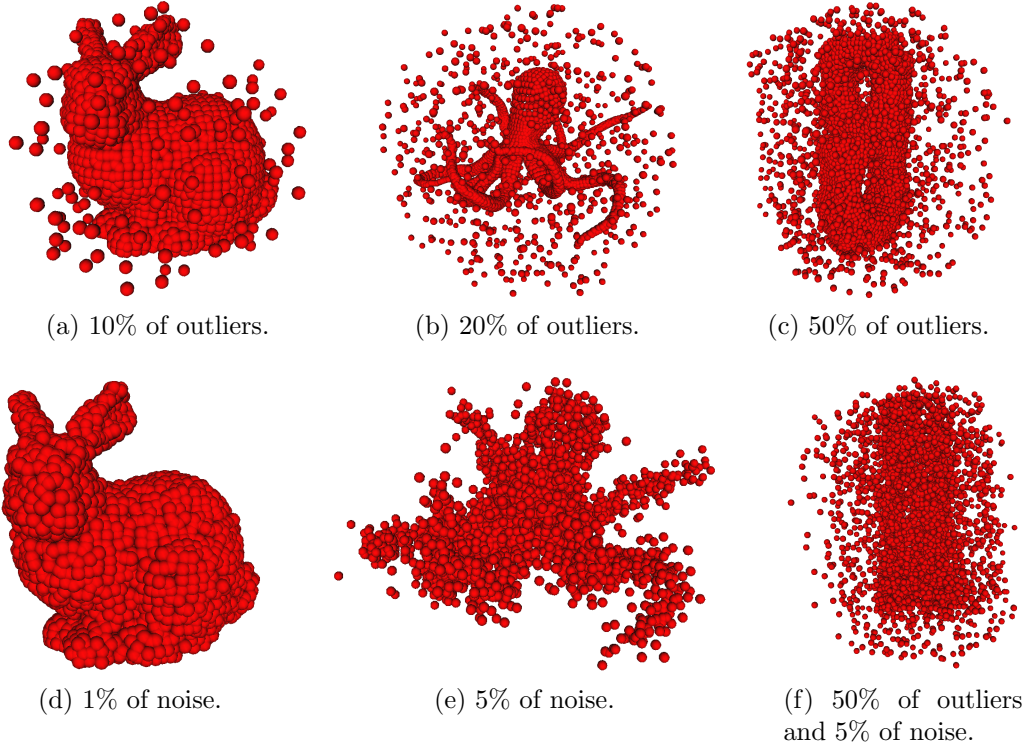
Figure 4.2: Examples of cases with outliers, noise, and the hardest case combining the greatest amounts of outliers and noise.

In the second group of experiments we evaluate different combinations of overlapping (β) and individual region (α) amounts. The proposed values for β show different levels of overlapping, varying from a large to a small common region. Table 4.1 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 non-overlapping 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 overlapping with each point cloud having 25% as remaining points. Figure 4.3 shows two examples of partial overlapping, one with high and one with low overlapping amount β.

(a) $\alpha = 12.5\%$ and $\beta = 75\%$.

(b) $\alpha = 12.5\%$ and $\beta = 25\%$.
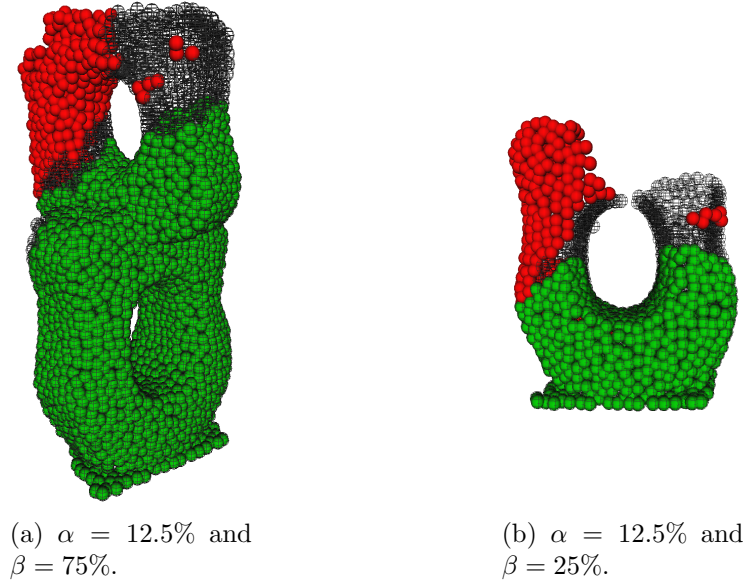
Figure 4.3: Two examples of partial overlapping. Red indicate points belonging only to the Model set, green are points belonging to the overlapped region, and black are points belonging only to the Data set.

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

Table 4.1: Percentage of Points of the Individual Region (α) and of the Overlapping Region (β) for the Second Group of Trials.

4.2   ERROR MEASUREMENT PROTOCOL

In the rigid registration literature we find many different measurements used to expose and compare the proposed algorithms. The most common are RMS (Root Mean Squared) error and computational time.

In this work we use the GTRMS error (Ground Truth Root Mean Squared error) (AIGER et al., 2008)Cejnog, 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 we 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 (CEJNOG, 2015), the number

of correct correspondences found by the 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 (**?**) used a threshold on the GTRMS and Labeled Error to define a success case. However, trials frequently have intermediate values, for registrations that have success on the coarse stage, but failed in the fine stage. Taking just the mean error with such trials might not express the correct performance of the method, and induce a wrong analysis. The same happens to 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 GTRMS range values comes from empirical observations. If an GTRMS error is higher than 0.5, it is set as 0.5. The score is a weighted mean of the histogram. The weighting matrix $\mathbf{W}$ has the same dimension of the histogram and 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 4.4 is an example of score for a trial, showing the histogram matrix with 4×4 bins and its weighting matrix.
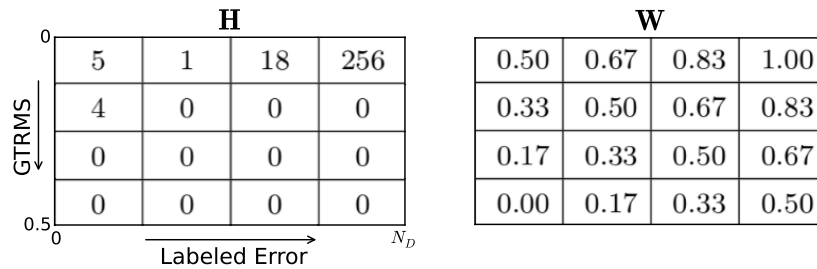
Figure 4.4: Example of a 4×4 bins histogram matrix $\mathbf{H}$ and a 4×4 weighting matrix $\mathbf{W}$. In this example the Score is: $\frac{1}{284}((5 \times 0.50) + (1 \times 0.67) + (\cdots)) = 0.9698$.

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

The score is normalized in the interval $[0, 1]$. However, the score is not a success rate. The labeled error is expected to decrease when noise is applied to the point cloud, since it is based on the matching function. Therefore, the score tends to be smaller in noise cases, but it does not mean the performance of the method is worse. The score is meant to be used to compare two methods in the same configuration of noise, outliers and overlapping amount. The diagram of Figure 4.5 summarizes the execution of our protocol.
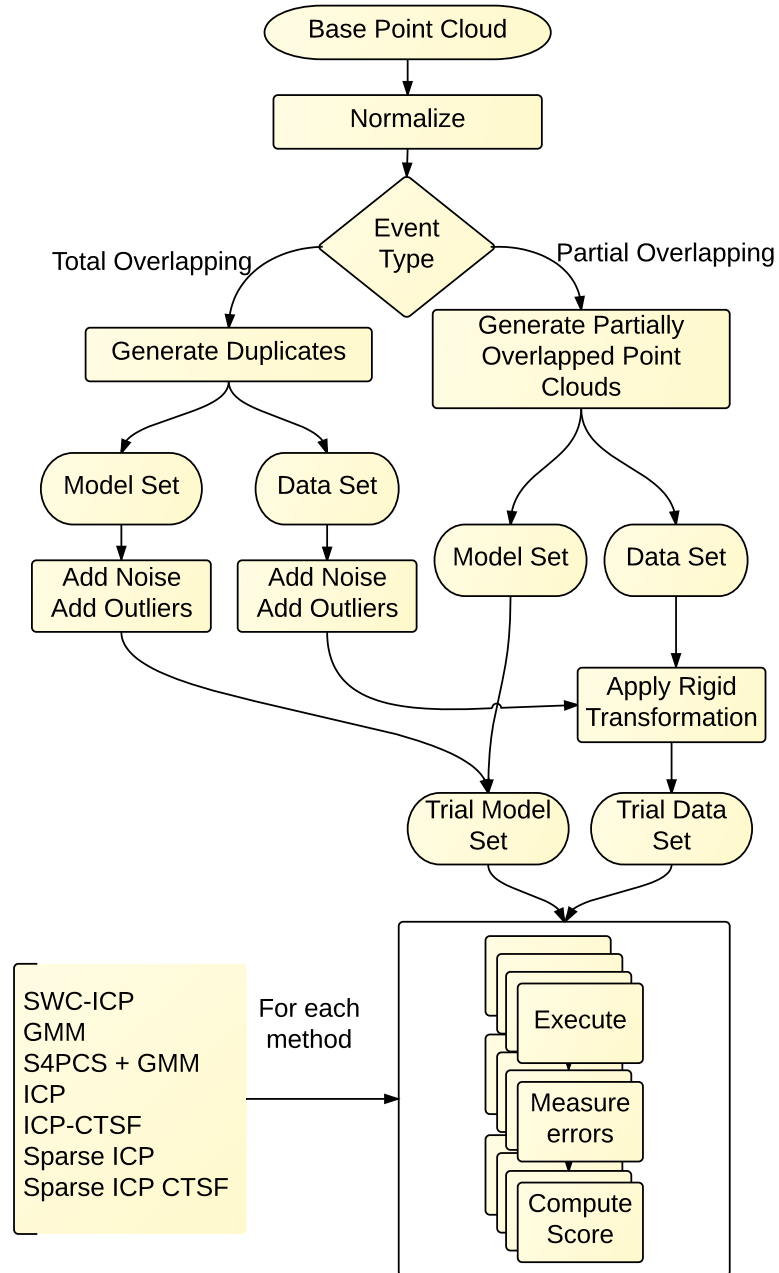
Figure 4.5: Flow chart of our evaluation protocol.

4.3 RESULTS

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

- The GMM framework (JIAN; VEMURI, 2011), 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)(MELLADO et al., 2014), 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 two versions of the S4PCS, one followed by the original ICP and the other followed by the GMM, to exemplify the behavior of a coarse method followed by a fine method.

- The ICP-CTSF (CEJNOG, 2015), that uses the CTSF to estimate the best corresponding neighbors in the matching step of the ICP.

- The Sparse ICP (BOUAZIZ et al., 2013), 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.

- We also compare with a variation of the Sparse ICP using the CTSF, like presented by Cejnog (2015), that uses the CTSF matching to estimate the corresponding set 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++.

4.3.1 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: $p = 0.4$, $\mu = 10.0$, $\alpha = 1.2$, $max_{\mu} = 10^5$, $max_{icp} = 100$, $max_{outer} = 100$, $max_{inner} = 1$, $stop = 10^{-4}$. The Sparse ICP with CTSF use $k = 100\%$. 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 for both methods, with and without the second voting field. The Sparse ICP with CTSF and the ICP-CTSF with the anisotropic voting field use $k = 100\%$, the best size according to Cejnog (2015). When only the isotropic voting field is used the best list size of the ICP-CTSF changes. We present the best value found in each point cloud. Results with tensors estimated using the two voting fields are identified with "S.", while "F." represent tensors estimated using only the first voting field. All the other methods use the best configuration presented by the authors and the default when not specified.

Method Name	Clean	Outliers	Noise	Noise + outliers	Overall
SWC-ICP S. $k = 100\%$	1.0000	0.9004	0.7299	0.6264	0.7433
SWC-ICP S. $k = 75\%$	1.0000	0.9449	0.7260	0.6635	0.7723
SWC-ICP S. $k = 50\%$	1.0000	0.9353	0.7168	0.6248	0.7490
SWC-ICP S. $k = 15\%$	1.0000	0.8063	0.6871	0.4282	0.6135
SWC-ICP S. $k = 5\%$	1.0000	0.7311	0.4692	0.2168	0.4527
SWC-ICP S. $k = 1\%$	1.0000	0.6265	0.1147	0.0496	0.2839
SWC-ICP F. $k = 100\%$	1.0000	0.9340	0.7311	0.6547	0.7660
SWC-ICP F. $k = 75\%$	1.0000	0.9472	0.7308	0.6709	0.7774
SWC-ICP F. $k = 50\%$	1.0000	0.9389	0.7293	0.6614	0.7703
SWC-ICP F. $k = 15\%$	1.0000	0.9165	0.7226	0.6122	0.7390
SWC-ICP F. $k = 5\%$	1.0000	0.8861	0.6978	0.5173	0.6798
SWC-ICP F. $k = 1\%$	1.0000	0.9358	0.3295	0.1148	0.4297
GMM	0.5370	0.5261	0.3985	0.3859	0.4356
S4PCS	0.9991	0.2125	0.6648	0.1244	0.3094
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 S. $k = 100\%$	1.0000	0.9351	0.7342	0.6561	0.7675
ICP-CTSF F. $k = 75\%$	1.0000	0.9824	0.7340	0.7048	0.8037
Sparse ICP	0.4610	0.3479	0.3260	0.2554	0.3074
Sparse ICP CTSF S.	1.0000	0.7370	0.6936	0.5202	0.6433
Sparse ICP CTSF F.	1.0000	0.6954	0.6329	0.4783	0.6018

Table 4.2: Combined score per case on the Bunny.

The Tables 4.2, 4.3, 4.4 and 4.5 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 average score. This overall value is our final score for the methods compared in the total overlap-

Method Name	Clean	Outliers	Noise	Noise + outliers	Overall
SWC-ICP S. $k = 100\%$	1.0000	0.8362	0.5735	0.5054	0.6407
SWC-ICP S. $k = 75\%$	1.0000	0.7908	0.5601	0.4221	0.5854
SWC-ICP S. $k = 50\%$	1.0000	0.7283	0.4932	0.3079	0.5016
SWC-ICP S. $k = 15\%$	1.0000	0.6526	0.4086	0.2347	0.4320
SWC-ICP S. $k = 5\%$	1.0000	0.7308	0.3233	0.1941	0.4170
SWC-ICP S. $k = 1\%$	1.0000	0.6953	0.1622	0.1057	0.3370
SWC-ICP F. $k = 100\%$	1.0000	0.7470	0.5767	0.4676	0.6000
SWC-ICP F. $k = 75\%$	1.0000	0.8476	0.5762	0.5135	0.6480
SWC-ICP F. $k = 50\%$	1.0000	0.8539	0.5757	0.5152	0.6503
SWC-ICP F. $k = 15\%$	1.0000	0.8257	0.5731	0.4532	0.6119
SWC-ICP F. $k = 5\%$	1.0000	0.7538	0.5626	0.3806	0.5559
SWC-ICP F. $k = 1\%$	1.0000	0.8602	0.3023	0.1731	0.4353
GMM	0.3793	0.3756	0.2519	0.2432	0.2891
S4PCS	0.9351	0.4066	0.5452	0.2693	0.4051
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 S. $k = 100\%$	1.0000	0.8824	0.5755	0.4803	0.6400
ICP-CTSF F. $k = 50\%$	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 S.	1.0000	0.9551	0.5711	0.5292	0.6819
Sparse ICP CTSF F.	1.0000	0.7626	0.5532	0.4187	0.5773

Table 4.3: Combined score per case on the Octopus.

Method Name	Clean	Outliers	Noise	Noise + outliers	Overall
SWC-ICP S. $k = 100\%$	1.0000	0.9208	0.6196	0.5353	0.6845
SWC-ICP S. $k = 75\%$	1.0000	0.7610	0.6205	0.4615	0.6077
SWC-ICP S. $k = 50\%$	1.0000	0.8367	0.6166	0.4895	0.6400
SWC-ICP S. $k = 15\%$	1.0000	0.7548	0.4888	0.3242	0.5156
SWC-ICP S. $k = 5\%$	1.0000	0.7032	0.3252	0.2145	0.4206
SWC-ICP S. $k = 1\%$	1.0000	0.8217	0.1961	0.1542	0.3986
SWC-ICP F. $k = 100\%$	1.0000	0.8100	0.6253	0.4698	0.6249
SWC-ICP F. $k = 75\%$	1.0000	0.9342	0.6353	0.5305	0.6880
SWC-ICP F. $k = 50\%$	1.0000	0.9533	0.6367	0.5631	0.7094
SWC-ICP F. $k = 15\%$	1.0000	0.9028	0.6097	0.4652	0.6433
SWC-ICP F. $k = 5\%$	1.0000	0.8797	0.5658	0.3930	0.5940
SWC-ICP F. $k = 1\%$	1.0000	0.9699	0.3053	0.1896	0.4715
GMM	0.5251	0.5195	0.3454	0.3510	0.4067
S4PCS	0.9928	0.5075	0.6100	0.3651	0.4938
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 S. $k = 100\%$	1.0000	0.9060	0.6333	0.5490	0.6899
ICP-CTSF F. $k = 50\%$	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 S.	1.0000	0.7074	0.4900	0.4365	0.5601
Sparse ICP CTSF F.	1.0000	0.8464	0.5557	0.5137	0.6444

Table 4.4: Combined score per case on the Happy Buddha.

ping test. The expanded tables with the results detailed by level of noise and outliers are presented in Tables C.1, C.2, C.3 and C.4, of the Appendix C.

Figures 4.6, 4.7, 4.8 and 4.9 show the score combining the clean point cloud, outliers, noise and outliers together with noise, detailed by angle. The Figures show the best SWC-ICP, the GMM, the Super 4PCS with GMM, the original ICP, the best ICP-CTSF,

Method Name	Clean	Outliers	Noise	Noise + outliers	Overall
SWC-ICP S. $k = 100\%$	1.0000	0.6544	0.6132	0.3928	0.5455
SWC-ICP S. $k = 75\%$	1.0000	0.7465	0.5652	0.4051	0.5667
SWC-ICP S. $k = 50\%$	1.0000	0.6126	0.5622	0.3394	0.4999
SWC-ICP S. $k = 15\%$	1.0000	0.7582	0.3862	0.2378	0.4561
SWC-ICP S. $k = 5\%$	1.0000	0.7282	0.2330	0.1464	0.3774
SWC-ICP S. $k = 1\%$	1.0000	0.7590	0.1594	0.1186	0.3589
SWC-ICP F. $k = 100\%$	1.0000	0.6281	0.5960	0.3654	0.5227
SWC-ICP F. $k = 75\%$	1.0000	0.8368	0.6292	0.4425	0.6189
SWC-ICP F. $k = 50\%$	1.0000	0.8845	0.6472	0.4864	0.6557
SWC-ICP F. $k = 15\%$	1.0000	0.8396	0.5891	0.3986	0.5909
SWC-ICP F. $k = 5\%$	1.0000	0.8473	0.4897	0.3383	0.5460
SWC-ICP F. $k = 1\%$	1.0000	0.9747	0.2441	0.1411	0.4387
GMM	0.5283	0.5383	0.3643	0.3594	0.4190
S4PCS	0.9835	0.4314	0.5351	0.2815	0.4198
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 S. $k = 100\%$	1.0000	0.7192	0.5597	0.4272	0.5701
ICP-CTSF F. $k = 50\%$	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 S.	0.9889	0.6286	0.4244	0.3950	0.5078
Sparse ICP CTSF F.	0.8904	0.5886	0.4161	0.3892	0.4853

Table 4.5: Combined score per case on the Genus-2.

the original Sparse ICP and the Sparse ICP with CTSF. The Super 4PCS with ICP and other values of k for the SWC-ICP and the ICP-CTSF are not presented because of their lower performance, improving the readability of the graphics.
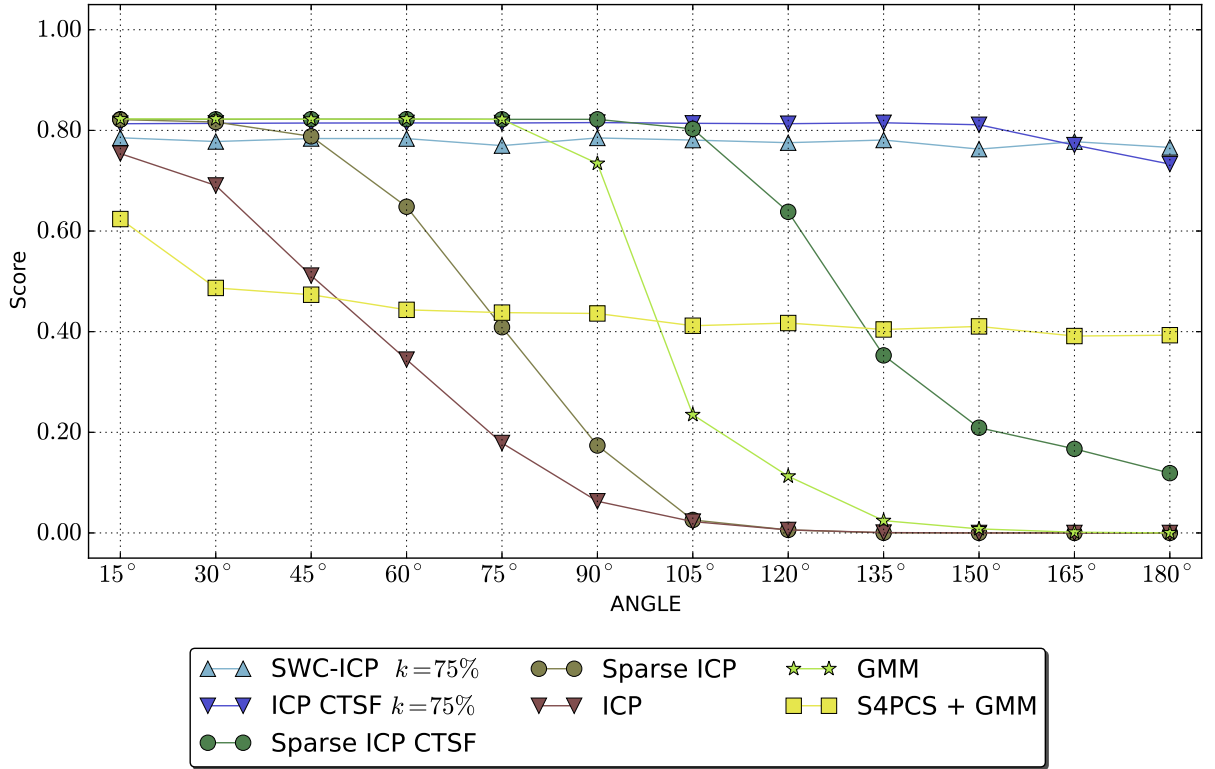
Figure 4.6: Overall score per angle on the Bunny.
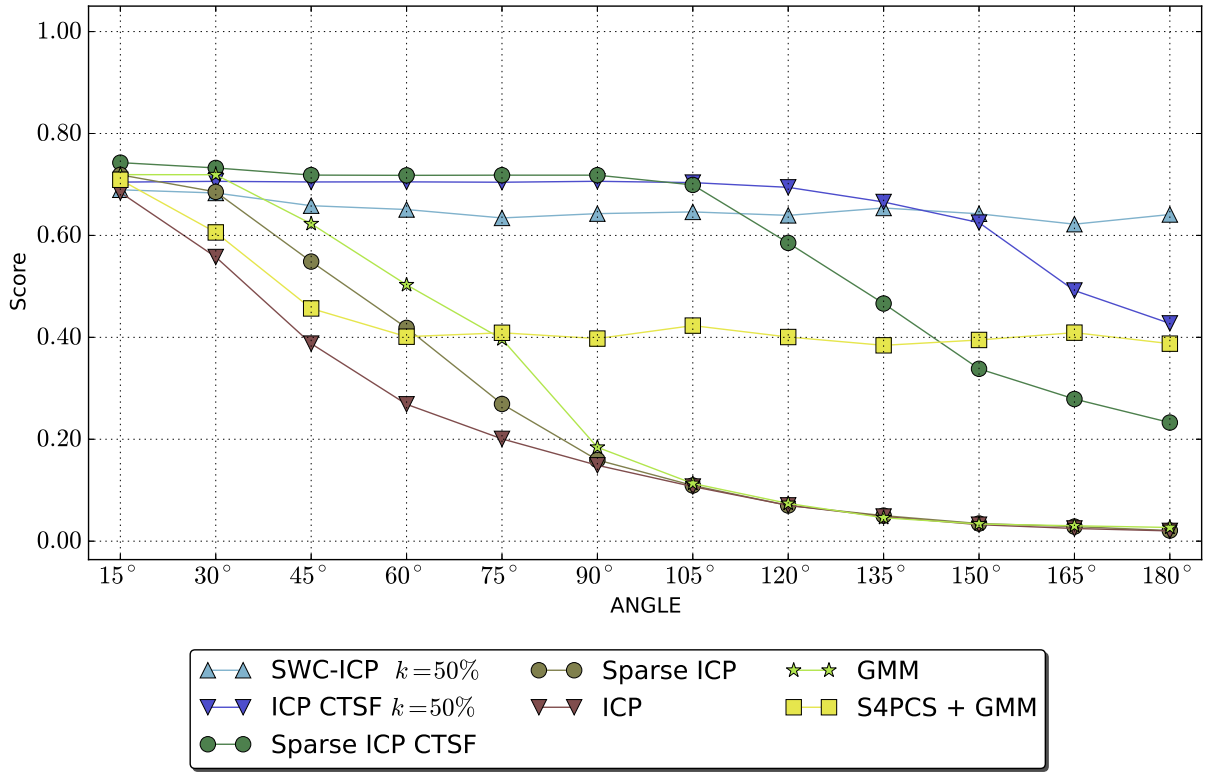
Figure 4.7: Overall score per angle on the Octopus.
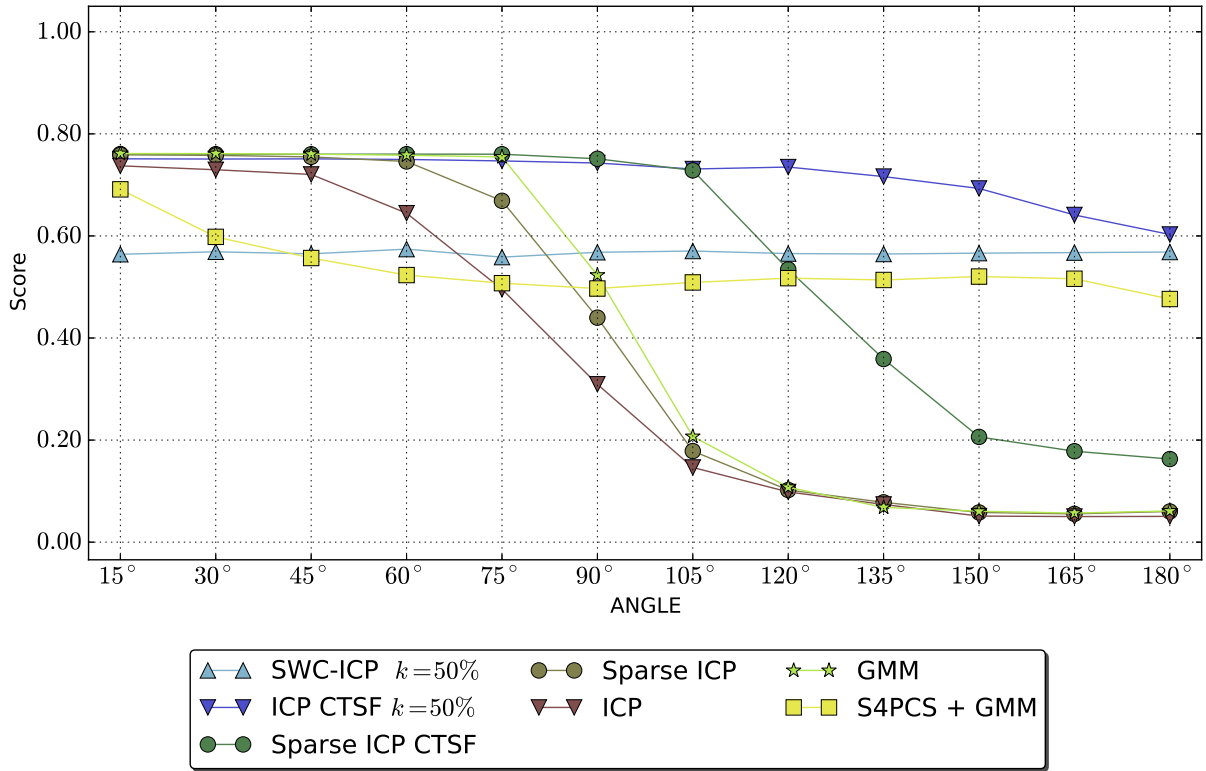
Figure 4.8: Overall score per angle on the Happy Buddha.
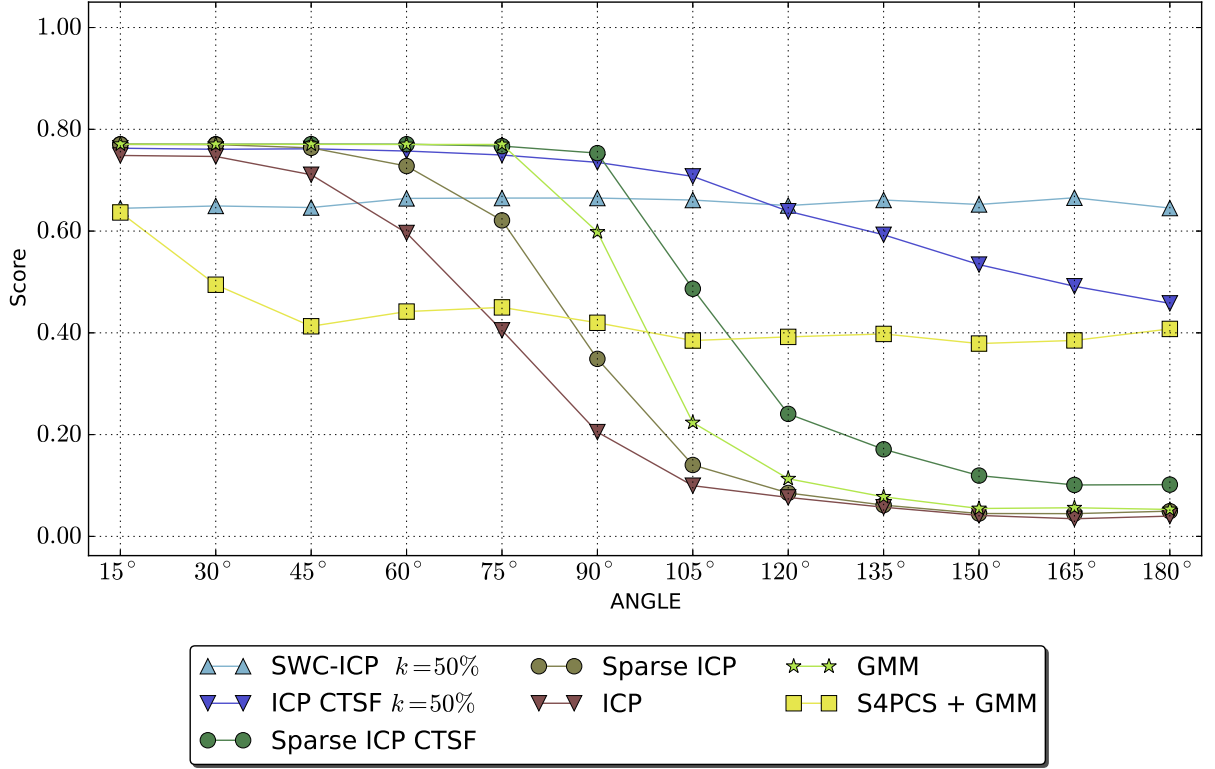
Figure 4.9: Overall score per angle on the Genus-2.

The results show that the Bunny was the easiest point cloud tested, with the overall scores higher than in other point clouds, for almost every method. That is justified because most of the surface of its body is smooth, the exceptions are the ears and the end of the paws. The other point clouds on the other way, presented harder scenarios. The difficulty on the Octopus comes from the high curvature of its tentacles. The geometry of the Genus-2 is mostly planar, but it has two symmetry axes which frequently lead the registration to false local optima. The Happy Buddha also have some degree of symmetry along its vertical axis, and differently from the Genus-2, it is more detailed and have just a few planar surfaces.

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, and using the first voting field by itself produced better results, especially on small neighbors list sizes k. The tensor voting process, used to estimate the tensors, is known for its robustness to outliers, confirmed with these results. Even with high amount of outliers, the structuring element was able to find the structure of the inliers, producing similar tensors for the same point on both

model and data point clouds.

The scores of the case with only noise are lower than those with only outliers, for almost every method. This behavior is expected, since noise affects directly the 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. Differently from the previous case, 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 19 points, which is too small to 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 comparisons between methods are similar to the case with just noise. The SWC-ICP with $k = 1\%$ is also the worst method. The reason is also the same, aggravated by the outliers. Again, using only the first voting field produced better results, and although small neighbors list sizes k still have lower performance, they doubled their score.

Figures 4.6, 4.7, 4.8 and 4.9 show that although the SWC-ICP has a slightly smaller score than the ICP-CTSF, in the Octopus and in the Genus-2 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 angles. 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 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 Figures that these three methods have even 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 a better method for wider angle situations. The addition of the CTSF to the Sparse ICP also proved to have very good results, comparable to both the SWC-ICP and the ICP-CTSF. It is clear

to see the improvement made by using only the first voting field, both on the SWC-ICP and on the ICP-CTSF. The performance gain on the ICP-CTSF was even greater than our presented method, and we recommend using it for all methods that benefits from the CTSF matching.

4.3.2 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 (2015). Four different neighbors list sizes were evaluated in the SWC-ICP, the two greatest and the 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 discard the amount α of the individual region size because in a real situations this value is unknown.

The Tables 4.6, 4.7, 4.8 and 4.9 show the scores detailed for all combinations of overlapping and individual regions presented on Table 4.1, 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 size, the better are the scores. This behavior is expected, since the number of points with correct correspondences is larger.

Bunny									
α	12.5%				25.0%				Overall
β	**12.5%**	**25.0%**	**50.0%**	**75.0%**	**12.5%**	**25.0%**	**37.5%**	**50.0%**	
SWC-ICP S. $k = 100\%$	0.1689	0.2446	0.4671	0.5237	0.1251	0.1421	0.2270	0.2579	0.2695
SWC-ICP S. $k = 75\%$	0.1762	0.2574	0.5562	0.8291	0.1547	0.1454	0.2523	0.3433	0.3394
SWC-ICP S. $k = 5\%$	0.2595	0.4177	0.7170	0.8802	0.1850	0.2418	0.3570	0.4855	0.4430
SWC-ICP S. $k = 1\%$	0.2709	0.4186	0.7178	0.8840	0.2177	0.2910	0.4218	0.5870	0.4761
SWC-ICP F. $k = 100\%$	0.1774	0.1719	0.2858	0.6557	0.1110	0.0679	0.0784	0.1797	0.2160
SWC-ICP F. $k = 75\%$	0.1832	0.2353	0.3612	0.7436	0.1074	0.0938	0.1365	0.1964	0.2572
SWC-ICP F. $k = 5\%$	0.3418	0.4043	0.5115	0.8158	0.2555	0.3356	0.4153	0.4928	0.4466
SWC-ICP F. $k = 1\%$	0.3432	0.4067	0.5160	0.8162	0.3070	0.3506	0.4217	0.4906	0.4565
GMM	0.1930	0.2203	0.3349	0.4566	0.1417	0.1297	0.1803	0.2611	0.2397
S4PCS	0.2324	0.3994	0.6918	0.8661	0.1308	0.2205	0.4355	0.6354	0.4515
S4PCS + GMM	0.2024	0.3420	0.6420	0.8689	0.1333	0.1446	0.3191	0.5278	0.3975
ICP	0.2140	0.1906	0.2223	0.2509	0.1655	0.1512	0.1578	0.1778	0.1912
ICP-CTSF S.	0.2350	0.3672	0.5587	0.6686	0.1918	0.2293	0.3177	0.4340	0.3753
ICP-CTSF F.	0.3459	0.4149	0.5340	0.8486	0.2891	0.3459	0.4197	0.5004	0.4623
Sparse ICP	0.2292	0.2550	0.3032	0.3692	0.1722	0.2031	0.2490	0.2912	0.2590
Sparse ICP CTSF S.	0.3184	0.5476	0.6997	0.8207	0.2421	0.4408	0.5744	0.6699	0.5392
Sparse ICP CTSF F.	0.4341	0.5780	0.7564	0.8846	0.3800	0.5498	0.6775	0.7493	0.6262

Table 4.6: Partial Overlapping Results - Bunny

Octopus									
α	12.5%				25.0%				Overall
β	**12.5%**	**25.0%**	**50.0%**	**75.0%**	**12.5%**	**25.0%**	**37.5%**	**50.0%**	
SWC-ICP S. $k = 100\%$	0.1610	0.1811	0.1897	0.3049	0.1442	0.1476	0.1618	0.1682	0.1823
SWC-ICP S. $k = 75\%$	0.1997	0.2172	0.2359	0.3630	0.1716	0.1604	0.1540	0.1506	0.2065
SWC-ICP S. $k = 5\%$	0.2810	0.3287	0.4444	0.7364	0.2363	0.2471	0.2536	0.3177	0.3557
SWC-ICP S. $k = 1\%$	0.2734	0.3549	0.5993	0.8763	0.2399	0.3208	0.3551	0.4257	0.4307
SWC-ICP F. $k = 100\%$	0.2844	0.2992	0.2603	0.3086	0.2364	0.2240	0.2231	0.2056	0.2552
SWC-ICP F. $k = 75\%$	0.3006	0.3188	0.3653	0.4457	0.2396	0.2434	0.2241	0.2166	0.2943
SWC-ICP F. $k = 5\%$	0.3987	0.4230	0.4548	0.5026	0.3043	0.3389	0.3434	0.3874	0.3941
SWC-ICP F. $k = 1\%$	0.4083	0.4320	0.4669	0.5006	0.3310	0.3556	0.3637	0.4213	0.4099
GMM	0.3156	0.3164	0.3397	0.3471	0.2622	0.2721	0.2885	0.2719	0.3017
S4PCS	0.4026	0.4390	0.3908	0.4493	0.2530	0.2681	0.2506	0.3026	0.3445
S4PCS + GMM	0.3649	0.3993	0.4146	0.5831	0.2554	0.2701	0.2637	0.2943	0.3557
ICP	0.2881	0.2870	0.2934	0.2864	0.2414	0.2504	0.2540	0.2438	0.2681
ICP-CTSF S.	0.3410	0.3934	0.6597	0.8522	0.2900	0.3470	0.3763	0.4209	0.4600
ICP-CTSF F.	0.4110	0.4279	0.4632	0.5009	0.3306	0.3487	0.3649	0.4204	0.4085
Sparse ICP	0.1326	0.2658	0.3504	0.3393	0.2285	0.2892	0.3370	0.3087	0.2814
Sparse ICP CTSF S.	0.1578	0.3816	0.6658	0.8743	0.2934	0.4819	0.6322	0.6763	0.5204
Sparse ICP CTSF F.	0.1644	0.4023	0.6796	0.8791	0.3220	0.5001	0.6334	0.7052	0.5347

Table 4.7: Partial Overlapping results - Octopus

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 same 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 4.10 shows an example of how the neighborhood gets different in points close to the edge of the overlapping region, as the number of

Happy Buddha									
α	12.5%				25.0%				Overall
β	**12.5%**	**25.0%**	**50.0%**	**75.0%**	**12.5%**	**25.0%**	**37.5%**	**50.0%**	
SWC-ICP S. $k = 100\%$	0.2620	0.2725	0.3594	0.7996	0.1861	0.2059	0.2390	0.3075	0.3290
SWC-ICP S. $k = 75\%$	0.2772	0.2887	0.3936	0.6329	0.2309	0.2077	0.2470	0.3033	0.3227
SWC-ICP S. $k = 5\%$	0.3061	0.4132	0.7087	0.8821	0.2314	0.2945	0.3925	0.5258	0.4693
SWC-ICP S. $k = 1\%$	0.3077	0.4159	0.7191	0.8821	0.2521	0.3166	0.4142	0.5342	0.4803
SWC-ICP F. $k = 100\%$	0.2703	0.2959	0.3179	0.6900	0.2124	0.2161	0.2598	0.3281	0.3238
SWC-ICP F. $k = 75\%$	0.2692	0.2917	0.3731	0.7071	0.2048	0.2309	0.2021	0.2742	0.3191
SWC-ICP F. $k = 5\%$	0.3546	0.4104	0.4745	0.7121	0.2955	0.3177	0.3855	0.4724	0.4278
SWC-ICP F. $k = 1\%$	0.3683	0.4142	0.4793	0.7155	0.3134	0.3359	0.3880	0.4722	0.4359
GMM	0.2919	0.3122	0.2977	0.3489	0.2168	0.2604	0.2848	0.2906	0.2879
S4PCS	0.3033	0.4091	0.4582	0.5590	0.2317	0.3051	0.3277	0.4161	0.3763
S4PCS + GMM	0.2925	0.3677	0.3968	0.5612	0.2255	0.2789	0.2829	0.3568	0.3453
ICP	0.3035	0.3003	0.3206	0.3489	0.2344	0.2206	0.2393	0.2649	0.2791
ICP-CTSF S.	0.3277	0.4398	0.6762	0.8723	0.2627	0.3233	0.3853	0.5022	0.4737
ICP-CTSF F.	0.3625	0.4189	0.4786	0.7498	0.3050	0.3352	0.3882	0.4746	0.4391
Sparse ICP	0.2406	0.3198	0.3955	0.4719	0.2321	0.2567	0.3174	0.3725	0.3258
Sparse ICP CTSF S.	0.2383	0.4898	0.7066	0.8355	0.2715	0.4489	0.6088	0.7129	0.5390
Sparse ICP CTSF F.	0.3097	0.5014	0.7466	0.8748	0.3308	0.5148	0.6488	0.7384	0.5831

Table 4.8: Partial Overlapping results - Happy Buddha

Genus-2								
α	12.5%				25.0%				Overall
β	12.5%	25.0%	50.0%	75.0%	12.5%	25.0%	37.5%	50.0%	
SWC-ICP S. $k=100\%$	0.2336	0.2267	0.2519	0.3741	0.1642	0.1348	0.1292	0.0885	0.2004
SWC-ICP S. $k=75\%$	0.2488	0.2348	0.3120	0.4738	0.1741	0.1882	0.2090	0.2156	0.2570
SWC-ICP S. $k=5\%$	0.3060	0.4418	0.7077	0.8798	0.2119	0.3463	0.4083	0.4973	0.4749
SWC-ICP S. $k=1\%$	0.3238	0.4651	0.7162	0.8762	0.2906	0.3863	0.4093	0.5143	0.4978
SWC-ICP F. $k=100\%$	0.2561	0.2449	0.3353	0.3374	0.1577	0.2020	0.2395	0.1372	0.2386
SWC-ICP F. $k=75\%$	0.2537	0.2458	0.4096	0.4822	0.1663	0.1798	0.2044	0.1361	0.2593
SWC-ICP F. $k=5\%$	0.3634	0.4333	0.5202	0.5672	0.3232	0.3873	0.3880	0.4374	0.4273
SWC-ICP F. $k=1\%$	0.3743	0.4423	0.5223	0.5808	0.3589	0.3903	0.3903	0.4381	0.4369
GMM	0.2823	0.2985	0.3458	0.4238	0.2081	0.2527	0.2787	0.2951	0.2981
S4PCS	0.3233	0.4505	0.4894	0.5134	0.1956	0.2746	0.3084	0.3586	0.3642
S4PCS + GMM	0.2848	0.3433	0.4313	0.5581	0.1939	0.2506	0.2708	0.3336	0.3333
ICP	0.2987	0.2937	0.3262	0.3472	0.2410	0.2409	0.2320	0.2496	0.2787
ICP-CTSF S.	0.3381	0.4422	0.6573	0.7460	0.2807	0.3393	0.3464	0.4179	0.4460
ICP-CTSF F.	0.3717	0.4426	0.5322	0.6729	0.3352	0.3832	0.3706	0.4194	0.4408
Sparse ICP	0.2371	0.3247	0.3999	0.4694	0.2455	0.2933	0.3400	0.3975	0.3384
Sparse ICP CTSF S.	0.2654	0.4940	0.7105	0.8210	0.3198	0.5038	0.6000	0.6730	0.5485
Sparse ICP CTSF F.	0.3208	0.5177	0.7419	0.8318	0.3549	0.5431	0.6667	0.7144	0.5856

Table 4.9: Partial Overlapping results - Genus-2

(a) 5 neighbors. (b) 10 neighbors. (c) 15 neighbors.
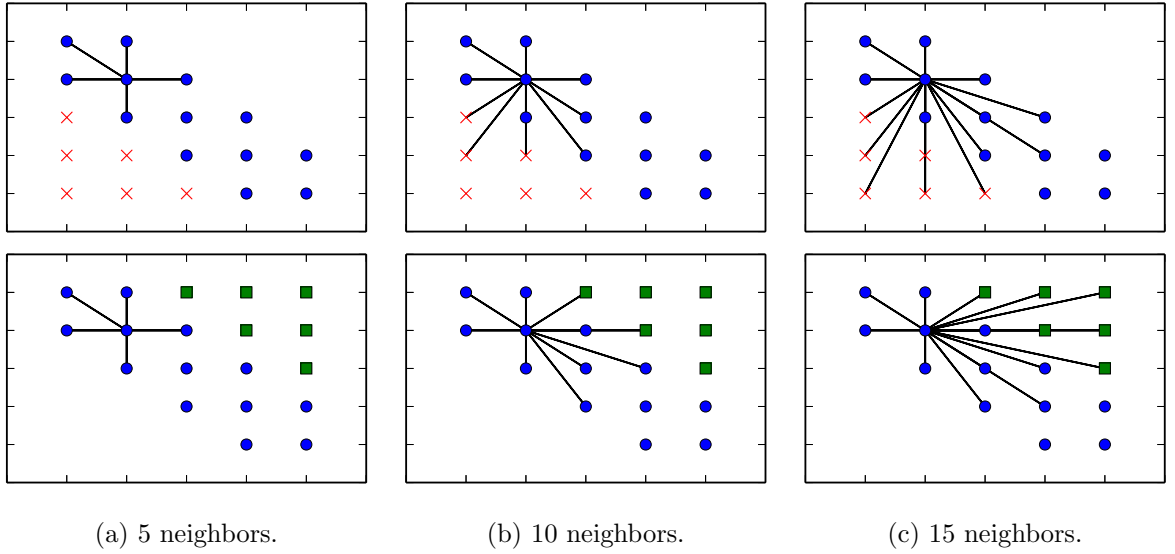
Figure 4.10: Examples of nearest neighbors for a given point in a case of partial overlapping. Blue points are inside the overlapping region, while red and green are in the unique region of each point cloud. Note that as the number of neighbors increase, the neighborhood considers points outside the overlapping region, which yields different tensors. Image from Cejnog (2015).

neighbors increase.

In this way, $k = 1\%$ is the best neighborhood size for the SWC-ICP with partial overlapping point clouds. Other overlapping combinations showed to be hard for all the methods tested, as the scores were mostly below 0.7. With smaller overlapping like 12.5% and 25%, the scores were even lower. This is 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.

The use of only the first voting field on partial overlapping point clouds does not have the same boost of performance. Although it improves the Sparse ICP with CTSF, it degrades the scores of the SWC-ICP and the ICP-CTSF. We believe that in the presence of noise and outliers the isotropic field should also be the best choice, like in the previous test. This result however, does not change our opinion that the anisotropic voting field should remain out of our method. The performance with this field is still lower than the Sparse ICP with 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 large angular displacement.

5   CONCLUSION

We presented the SWC-ICP, an improvement of the Iterative Closest Point algorithm that benefits from the local geometry surrounding the points, encoded in second-order tensors. The method boosts the chance to converge to the correct local optimum, even in the presence of noise, outliers and partial overlapping, regardless of the initial pose. The main parameter of the method is the size of the list of neighbors k. 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 choice when both point clouds are the same.

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. Our coarse-to-fine approach was better in all cases than the Super 4PCS followed by an execution of the GMM framework. This setup simulates a process where a coarse method is applied first to give a better initial pose to a fine method.

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 our method is invariant to the initial angular displacement. Noise and outliers does not alter this behavior, although they degrade the accuracy of the registration.

We evaluated the impact of using only the isotropic voting field in the rigid registration problem, instead of using the two fields of the previous approach. In the presence of noise and outliers the first voting field yields more distinguishable tensors, which improves the quality of the CTSF matching, reducing the likeliness of wrong matches. As results indicate, for partial overlapping point clouds, without noise and outliers, the two voting fields are still a better approach. However, we point out that for real applications the point clouds usually contain some noise and outliers, situations where the first voting field is better. Furthermore, the performance gain of the anisotropic voting field in partial overlapping cases is still not enough to outperform the Sparse ICP with CTSF, the best method at the moment in this case.

A drawback of our method is the need of a preprocessing stage to compute the tensors,

that might be computationally expensive if the point cloud is too large, although parallelism exploitation in this step is possible. The second drawback is the performance with low overlapping regions of point clouds. The method requires over 50% of overlapping and less than 25% as individual region to show a reasonable result.

5.1 FUTURE WORKS

We believe that better techniques to estimate the tensors may improve even more the scores of the SWC-ICP. The current CTSF matching is based only on the shape of the tensors, and their relative orientation is not used. Additional voting fields can be applied instead, or after, the anisotropic voting field not used in our method. These fields could be designed to estimate other geometric features to improve even more the matching step of the SWC-ICP, yielding better results.

The presented evaluation protocol can still be extended. More models with different geometric features, and other measurements, like time or rotational error, can be included to strengthen the comparisons.

REFERENCES

AIGER, D.; MITRA, N. J.; COHEN-OR, D. 4-points congruent sets for robust pairwise surface registration. In: ACM. **ACM Transactions on Graphics (TOG)**, 2008. v. 27, n. 3, p. 85.

ALBARELLI, A.; RODOLÀ, E.; TORSELLO, A. Fast and accurate surface alignment through an isometry-enforcing game. **Pattern Recognition**, Elsevier, v. 48, n. 7, p. 2209–2226, 2015.

BESL, P.; MCKAY, N. A method for registration of 3-d shapes. **IEEE transactions on pattern analysis and machine intelligence**, IEEE Computer Society, v. 14, n. 2, p. 239–256, 1992.

BOUAZIZ, S.; TAGLIASACCHI, A.; PAULY, M. Sparse iterative closest point. **Computer Graphics Forum**, Blackwell Publishing Ltd, v. 32, n. 5, p. 113–123, 2013. ISSN 1467-8659.

CEJNOG, L. W. X. **Rigid registration based on local geometric dissimilarity**. Dissertação (Mestrado) — Post-Graduation Program on Computer Science, Federal University of Juiz de Fora, Juiz de Fora, MG, Brazil, 2015.

CHEN, C.-S.; HUNG, Y.-P.; CHENG, J.-B. Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, v. 21, n. 11, p. 1229–1234, 1999.

CORSINI, M.; CIGNONI, P.; SCOPIGNO, R. Efficient and flexible sampling with blue noise properties of triangular meshes. **Visualization and Computer Graphics, IEEE Transactions on**, IEEE, v. 18, n. 6, p. 914–924, 2012.

DÍEZ, Y.; ROURE, F.; LLADÓ, X.; SALVI, J. A qualitative review on 3d coarse registration methods. **ACM Computing Surveys (CSUR)**, ACM, v. 47, n. 3, p. 45, 2015.

DONOSER, M.; BISCHOF, H. 3d segmentation by maximally stable volumes (msvs). In: IEEE. **Pattern Recognition, 2006. ICPR 2006. 18th International Conference on**, 2006. v. 1, p. 63–66.

EGGERT, D. W.; LORUSSO, A.; FISHER, R. B. Estimating 3-d rigid body transformations: a comparison of four major algorithms. **Machine Vision and Applications**, Springer, v. 9, n. 5-6, p. 272–290, 1997.

FITZGIBBON, A. W. Robust registration of 2d and 3d point sets. **Image and Vision Computing**, Elsevier, v. 21, n. 13, p. 1145–1153, 2003.

GELFAND, N.; MITRA, N. J.; GUIBAS, L. J.; POTTMANN, H. Robust global registration. In: **Symposium on geometry processing**, 2005. v. 2, n. 3, p. 5.

GODIN, G.; RIOUX, M.; BARIBEAU, R. Three-dimensional registration using range and intensity information. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Photonics for Industrial Applications**, 1994. p. 279–290.

HORN, B. K. Closed-form solution of absolute orientation using unit quaternions. **JOSA A**, Optical Society of America, v. 4, n. 4, p. 629–642, 1987.

JIAN, B.; VEMURI, B. C. Robust point set registration using gaussian mixture models. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, v. 33, n. 8, p. 1633–1645, 2011.

JOHNSON, A. E.; HEBERT, M. Using spin images for efficient object recognition in cluttered 3d scenes. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, v. 21, n. 5, p. 433–449, 1999.

KINDLMANN, G. Superquadric tensor glyphs. In: EUROGRAPHICS ASSOCIATION. **Proceedings of the Sixth Joint Eurographics-IEEE TCVG conference on Visualization**, 2004. p. 147–154.

MAIER-HEIN, L.; FRANZ, A. M.; SANTOS, T. R. dos; SCHMIDT, M.; FANGERAU, M.; MEINZER, H.; FITZPATRICK, J. M. Convergent iterative closest-point algorithm to accomodate anisotropic and inhomogenous localization error. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, v. 34, n. 8, p. 1520–1532, 2012.

MAVRIDIS, P.; ANDREADIS, A.; PAPAIOANNOU, G. Efficient sparse icp. **Computer Aided Geometric Design**, Elsevier, v. 35, p. 16–26, 2015.

MEDIONI, G.; TANG, C.-K.; LEE, M.-S. Tensor voting: Theory and applications. **Proceedings of RFIA, Paris, France**, v. 3, 2000.

MELLADO, N.; AIGER, D.; MITRA, N. J. Super 4pcs fast global pointcloud registration via smart indexing. In: WILEY ONLINE LIBRARY. **Computer Graphics Forum**, 2014. v. 33, n. 5, p. 205–215.

MIAN, A.; BENNAMOUN, M.; OWENS, R. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. **International Journal of Computer Vision**, Springer, v. 89, n. 2-3, p. 348–361, 2010.

MORDOHAI, P.; MEDIONI, G. Tensor voting: a perceptual organization approach to computer vision and machine learning. **Synthesis Lectures on Image, Video, and Multimedia Processing**, Morgan & Claypool Publishers, v. 2, n. 1, p. 1–136, 2006.

OVSJANIKOV, M.; MÉRIGOT, Q.; MÉMOLI, F.; GUIBAS, L. One point isometric matching with the heat kernel. In: WILEY ONLINE LIBRARY. **Computer Graphics Forum**, 2010. v. 29, n. 5, p. 1555–1564.

POTTMANN, H.; WALLNER, J.; HUANG, Q.-X.; YANG, Y.-L. Integral invariants for robust geometry processing. **Computer Aided Geometric Design**, Elsevier, v. 26, n. 1, p. 37–60, 2009.

REYES, L.; MEDIONI, G.; BAYRO, E. Registration of 3d points using geometric algebra and tensor voting. **International Journal of Computer Vision**, Springer, v. 75, n. 3, p. 351–369, 2007.

RUSINKIEWICZ, S.; LEVOY, M. Efficient variants of the icp algorithm. In: IEEE. **3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on**, 2001. p. 145–152.

SALVI, J.; MATABOSCH, C.; FOFI, D.; FOREST, J. A review of recent range image registration methods with accuracy evaluation. **Image and Vision Computing**, Elsevier, v. 25, n. 5, p. 578–596, 2007.

SEGAL, A.; HAEHNEL, D.; THRUN, S. Generalized-icp. In: **Robotics: Science and Systems**, 2009. v. 2, n. 4.

SERVOS, J.; WASLANDER, S. L. Multi channel generalized-icp. In: IEEE. **Robotics and Automation (ICRA), 2014 IEEE International Conference on**, 2014. p. 3644–3649.

SHARP, G. C.; LEE, S. W.; WEHE, D. E. Invariant features and the registration of rigid bodies. In: IEEE. **Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on**, 1999. v. 2, p. 932–937.

SHARP, G. C.; LEE, S. W.; WEHE, D. K. Icp registration using invariant features. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, v. 24, n. 1, p. 90–102, 2002.

SUN, J.; OVSJANIKOV, M.; GUIBAS, L. A concise and provably informative multi-scale signature based on heat diffusion. In: WILEY ONLINE LIBRARY. **Computer Graphics Forum**, 2009. v. 28, n. 5, p. 1383–1392.

TOMBARI, F.; SALTI, S.; STEFANO, L. D. Unique signatures of histograms for local surface description. In: **Computer Vision–ECCV 2010**, 2010. p. 356–369.

VIEIRA, M. B.; MARTINS, P.; ARAUJO, A.; CORD, M.; PHILIPP-FOLIGUET, S. Smooth surface reconstruction using tensor fields as structuring elements. In: WILEY ONLINE LIBRARY. **Computer Graphics Forum**, 2004. v. 23, n. 4, p. 813–823.

WESTIN, C.-F.; PELED, S.; GUDBJARTSSON, H.; KIKINIS, R.; JOLESZ, F. A. Geometrical diffusion measures for mri from tensor basis analysis. **ISMRMÂŠ97**, p. 1742, 1997.

YANG, C.; MEDIONI, G. Object modelling by registration of multiple range images. **Image and vision computing**, Elsevier, v. 10, n. 3, p. 145–155, 1992.

YANG, J.; LI, H.; JIA, Y. Go-icp: Solving 3d registration efficiently and globally optimally. In: IEEE. **Computer Vision (ICCV), 2013 IEEE International Conference on**, 2013. p. 1457–1464.

ZAHARESCU, A.; BOYER, E.; VARANASI, K.; HORAUD, R. Surface feature detection and description with applications to mesh matching. In: IEEE. **Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on**, 2009. p. 373–380.

ZHONG, Y. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In: IEEE. **Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on**, 2009. p. 689–696.

Appendix A - ANISOTROPIC VOTING FIELD

The tensors $\mathbf{T}_p$ obtained in the first voting field are refined in the second voting field. A different structuring element that enhances coplanar structures is applied. This is done modifying the vote vector and the weighting function, ensuring that points aligned to the tangent plane have more influence. A metric based on the distance over an ellipsoid tangent to the points is used as attenuation factor, measuring the coplanarity of points and assuring that both the Euclidean distance and the low curvature of elliptical segments are favored. Unlike the first voting field, here each point p casts the vote on the neighbors q. These votes are also stick tensors formed by the product $\widehat{pq} \cdot \widehat{pq}^T$.

First, for all neighbors $q \in L(p)$ we align the canonical axes $\widehat{x}, \widehat{y}, \widehat{z}$ to the eigenvectors $(\widehat{e}_{1p}, \widehat{e}_{2p}, \widehat{e}_{3p})$ of $\mathbf{T}_p$. The transformation of q to this system is done applying the rotation matrix $\mathbf{R}_p = [\widehat{e}_{1p}, \widehat{e}_{2p}, \widehat{e}_{3p}]^T$ on the vector $\overrightarrow{pq}$. We denote p' the origin and q' the point q in this system.

$$q' = \mathbf{R}_p \cdot (\overrightarrow{pq}).$$

In spherical coordinates, the angle between $\overrightarrow{pq}$ and the plane defined by $\overrightarrow{e_{3p}}$ is trivial to find. Then, q' in spherical coordinates is:

$$\begin{cases} \rho_{q'} = \sqrt{q'^2_x + q'^2_y + q'^2_z}, \\ \theta_{q'} = \tan^{-1} \dfrac{q'_y}{q'_x}, \\ \phi_{q'} = \tan^{-1} \dfrac{q'_z}{\sqrt{q'^2_x + q'^2_y}}. \end{cases}$$

For each neighbor q' there is an ellipsoid E centered over the $\widehat{z'} - axis$ and tangent to p' and q'. The coplanarity between p and a neighbor q is proportional to the distance d_e between p' and q' over this ellipsoid, which is given by:

$$d_e(p, q) = \rho_{q'} \cos \phi_{q'} \left(1 + \left(2 - \frac{1}{\tan \alpha_{ellip}^2} \right) \tan^2 \phi_{q'} \right)^{\frac{\tan \alpha_{ellip}^2}{\cdot 2 \tan \alpha_{ellip}^2 - 1}}. \qquad (A.1)$$

The parameter α_{ellip} is proportional to the eccentricity of E, and controls the influence

of the coplanarity over the Euclidean distance.

From the spherical distance defined in Equation A.1 and the standard deviation σ_p from Equation 2.1.1, we define the influence f_{pq}, exerted from p on each neighbor $q \in L_k(p)$. Like in previous step, the farthest point in $L_k(p)$, using Euclidean distance, has influence 0.01.

The influence of points misaligned to the tangent plane defined by the normal of p', i.e. $\overrightarrow{e_3}_p$, is constrained by considering points with $\tan \phi_{q'} \leq \tan \phi_{max}$. Wider angles ϕ_{max} produce smoother results, while narrower angles allow more details at the cost of robustness to outliers.

$$f_{pq} = \begin{cases} e^{\dfrac{-d_e(p,q)}{\sigma_p^2}} & , \tan \phi_{q'} \leq \tan \phi_{max} \\[2ex] 0.0 & , \tan \phi_{q'} > \tan \phi_{max} \end{cases}$$

The vote vector $\widehat{v}_{pq}$ requires the computation of the angle $\beta_{q'}$, which is the angle formed by the x'-axis and the line tangent to the ellipsoid E at the point q'. In function of g and $\phi_{q'}$, $\beta_{q'}$ is:

$$\beta_{q'} = \tan^{-1} \frac{2g^2 \tan \phi_{q'}}{g^2 - \tan^2 \phi_{q'}}$$

By the replacement of the angle $\phi_{q'}$ for $\beta_{q'}$ and the conversion of q' back to Euclidean coordinates, we obtain a vector $\widehat{v'}_{pq}$.

$$\widehat{v'}_{pq} = (\cos \theta_{q'} \cos \beta_{q'})\widehat{i} + (\sin \theta_{q'} \cos \beta_{q'})\widehat{j} + (\sin \beta_{q'})\widehat{k}. \tag{A.2}$$

The vote vector $\widehat{v}_{pq}$ is obtained applying the inverse rotation matrix $\mathbf{R}_p^{-1}$:

$$v_{pq} = \mathbf{R}_p^{-1} \cdot v'_{pq}$$

Figure A.1 shows a cut of the plan $y' = 0$ depicting the vector $\widehat{v}_{pq}$ and the angles $\phi_{q'}$ and $\beta_{q'}$ for an arbitrary point, in a case where $\alpha_{ellip} = 30°$.

Finally, the resulting tensor $\mathbf{S}_p$ for a point p is composed by the weighted sum of the tensors built from the votes received on the point, cast by all the points that have p as a neighbor:

$$\mathbf{S}_p = \sum_{q \in L_k^{-1}(p)} f_{qp} \cdot v_{qp} \cdot v_{qp}^T \tag{A.3}$$
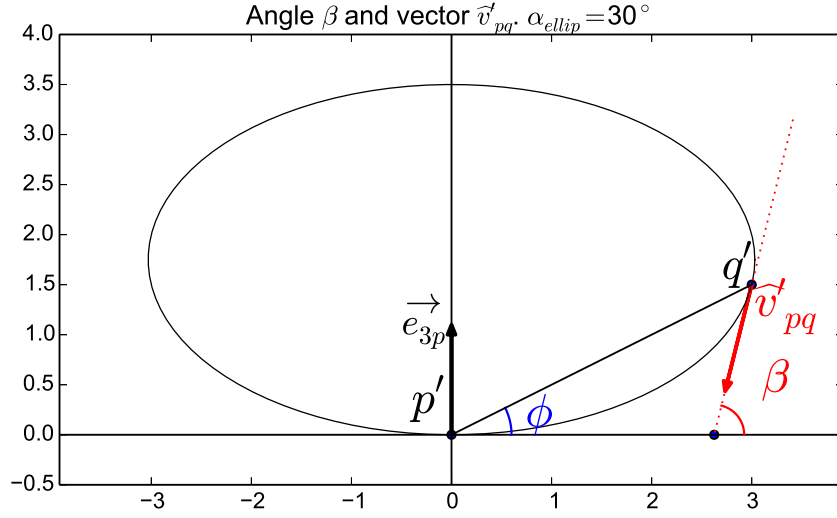
Figure A.1: $2D$ geometric representation of the angles ϕ, β and vector v_{pq} of an arbitrary point q'. Note that v_{pq} is normalized.

The second voting field can be applied multiple times, in order to enforce even more the planar shapes of the tensors. In Cejnog (2015) it is applied until the mean c_p stops increasing or 100 reapplications are made.

(a) k=1%. (b) k=5%. (c) k=10%. (d) k=50%. (e) k=75%. (f) k=100%.
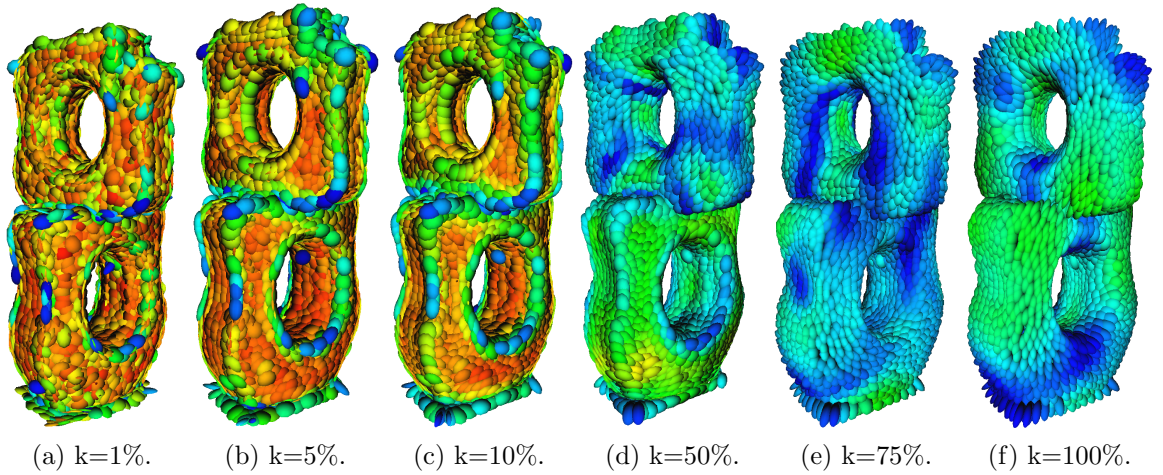
Figure A.2: Variation of the nearest neighbors list size using the second voting field. Reddish colors represent tensors with high c_p, bluish colors in contrast, represent tensors with low c_p.

Figure A.2 shows the final tensor estimation varying the neighbors list size k after the iterative application of the second voting field process, on the Genus-2. We note a more reddish point cloud using $k = 1\%$, $k = 5\%$ and $k = 10\%$. In these cases, since the Genus-2 has many planar regions there is a higher chance that the neighbors are coplanar with the point, resulting in a planar tensor. Conversely, with a larger list, neighbors far from the

tangent plane of the point will influence the tensor, resulting in a less planar shape, like Figures A.2 (d), (e) and (f) show. Due the limit angle ϕ_{max}, even with $k = 100\%$, not all neighbors contribute to the estimation. Although the tensors on the top and bottom of Figure A.2 (f) are still pointed to the center, their shape is less of a stick then those on the top and bottom of Figure 2.3 (f).

Appendix B - PSEUDOCODES

The Algorithm 1 shows the main algorithm of the SWC-ICP. The differences of the original method are the estimation of the tensors for both point clouds, needed to estimate the correspondences based on the CTSF, and weighting strategy updated only when a local optima is reached.

Algorithm 1: SWC-ICP

Data: M, D, k, b;

begin

    $computeNeighborsList(M, k)$;

    $computeNeighborsList(D, k)$;

    $tensorEstimation(M)$;

    $tensorEstimation(D)$;

    $S \leftarrow N_{CTSF}(M, D)$;

    $previousError \leftarrow INF$;

    $error \leftarrow INF$;

    $w \leftarrow 10^5$;

    while $w > 10^{-6}$ **do**

        $E \leftarrow N_E(M, D)$;

        $T \leftarrow transformationEstimation(D, E, S, w)$;

        $previousError \leftarrow error$;

        $error \leftarrow RMS(T \cdot D, M)$;

        if $error < previousError$ **then**

            $applyTransformation(D, T)$;

        else

            $error \leftarrow previousError$;

            $w \leftarrow w * b$;

        end if

    end while

end

The core of the SWC-ICP includes the correspondences set S, based on the CTSF. Additionally, the centroid $\overrightarrow{\mu_S}$, the cross-covariance matrix Σ_{DS} and the final covariance Σ_{DES} are different from the original ICP.

Algorithm 2: Transformation Estimation

Data: D, E, S, w;

begin

    $\overrightarrow{\mu_D} \leftarrow getCentroid(D)$;

    $\overrightarrow{\mu_E} \leftarrow getCentroid(E)$;

    $\overrightarrow{\mu_S} \leftarrow getCentroid(S)$;

    $\Sigma_{DE} \leftarrow computeCovariance(D, E, \overrightarrow{\mu_D}, \overrightarrow{\mu_E})$;

    $\Sigma_{DS} \leftarrow computeCovariance(D, S, \overrightarrow{\mu_D}, \overrightarrow{\mu_S})$;

    $\Sigma_{DES} \leftarrow \Sigma_{DE} + w * \Sigma_{DS}$;

    $N \leftarrow getQuaternionMatrix(\Sigma_{DES})$;

    $\mathbf{R} \leftarrow getGreatestEigenVector(N)$;

    $\overrightarrow{\mathbf{t}} \leftarrow \overrightarrow{\mu_E} - \mathbf{R} * \overrightarrow{\mu_D}$

    return $T(\mathbf{R}, \overrightarrow{\mathbf{t}})$;

end

The Algorithm 3 shows the tensor estimation process, with a straightforward implementation, given the formulation of the Equations 2.1.1 and 2.1.

Algorithm 3: Tensor Estimation - First Voting Field

Data: P;

begin

    for $p \in P$ **do**

        $\mathbf{T}_p \leftarrow zeroMatrix(3,3)$;

        $q_f \leftarrow getFarthestPoint(P,p)$;

        $\sigma_p \leftarrow \sqrt{\frac{||\overrightarrow{pq_f}||^2}{\ln 0.01}}$;

        for $q \in L_k(p)$ **do**

            $g \leftarrow e^{\frac{-||\overrightarrow{pq}||^2}{\sigma_p^2}}$;

            $\mathbf{T}_p \leftarrow \mathbf{T}_p + g \cdot \widehat{pq} \cdot \widehat{pq}^T$;

        end for

    end for

end

The Algorithm 4 shows the second voting field used in the original ICP-CTSF. The parameters are the point cloud P, the angle α_{ellip} that controls the eccentricity of the ellipsoid, and the limit angle ϕ_{max}. For a detailed discussion of how these two angles impact the ICP see Cejnog's dissertation Cejnog (2015).

Algorithm 4: Tensor Estimation - Second Voting Field

Data: P, $\alpha_{ellip} \leftarrow \frac{\pi}{4}$, $\phi_{max} \leftarrow \frac{\pi}{4}$;

begin

    for $p \in P$ **do**

        $\mathbf{S}_p \leftarrow zeroMatrix(3,3)$;

        $q_f \leftarrow getFarthestPoint(P,p)$;

        $\sigma_p \leftarrow \sqrt{\frac{||\overrightarrow{pq_f}||^2}{\ln 0.01}}$;

        $\widehat{e}_1, \widehat{e}_2, \widehat{e}_3 \leftarrow eigenDecomposition(\mathbf{T}_p)$;

        $\mathbf{R}_p \leftarrow [\widehat{e}_{1x}, \widehat{e}_{1y}, \widehat{e}_{1z}, \widehat{e}_{2x}, \widehat{e}_{2y}, \widehat{e}_{2z}, \widehat{e}_{3x}, \widehat{e}_{3y}, \widehat{e}_{3z}]$;

        for $q \in L_k(p)$ **do**

            /* Align to the eigensystem of p */

            $q' \leftarrow \mathbf{R}_p \cdot \overrightarrow{pq}$;

            /* Spherical Coordinates   */

            $\rho_{q'} \leftarrow \sqrt{{q'_x}^2 + {q'_y}^2 + {q'_z}^2}$;

            $\phi_{q'} \leftarrow \tan^{-1} \frac{q'_z}{\sqrt{{q'_x}^2+{q'_y}^2}}$;

            $\theta_{q'} \leftarrow \tan^{-1} \frac{q'_y}{q'_x}$;

            /* Compute vote vector   */

            $g \leftarrow \tan \alpha_{ellip}$;

            $\beta_{q'} \leftarrow \tan^{-1} \frac{2g^2 \tan \phi_{q'}}{g^2 - \tan^2 \phi_{q'}}$;

            $\widehat{v'}_x \leftarrow \cos \theta_{q'} \cos \beta_{q'}$;

            $\widehat{v'}_y \leftarrow \sin \theta_{q'} \cos \beta_{q'}$;

            $\widehat{v'}_z \leftarrow \sin \beta_{q'}$;

            /* Distance over ellipsoid   */

            $d_e(p,q) \leftarrow \rho_{q'} \cos \phi_{q'} (1.0 + (2.0 - (\frac{1.0}{g^2})) \tan^2 \phi_{q'})^{\frac{g^2}{2g^2-1}}$;

            $f_{pq} \leftarrow e^{\frac{-d_e(p,q)^2}{\sigma_p}}$;

            /* Rotate back   */

            $\widehat{v}_{pq} \leftarrow \mathbf{R}_p^{-1} \cdot \widehat{v'}_{pq}$;

            if $(|\tan \phi_{q'}| \leq |\tan \phi_{max}|)$ **then** /* Checks the limit angle   */

                $\mathbf{S}_p \leftarrow \mathbf{S}_p + f_{pq} \cdot \widehat{v}_{pq} \cdot \widehat{v}_{pq}^T$;

            end if

        end for

    end for

end

Appendix C - DETAILED RESULTS

The Tables C.1, C.2, C.3 and C.4 show results combining all twelve angles, detailing each combination of noise and outlier amount.

SWC-ICP $k = 100\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.9668	0.9578	0.8774	0.9505
$\sigma = 0.01$	0.9445	0.9030	0.8806	0.8080	0.8840
$\sigma = 0.05$	0.5178	0.4782	0.4623	0.3963	0.4636
Overall	0.8208	0.7827	0.7669	0.6939	0.7660
SWC-ICP $k = 75\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.9684	0.9546	0.9187	0.9604
$\sigma = 0.01$	0.9450	0.8954	0.8898	0.8260	0.8890
$\sigma = 0.05$	0.5166	0.4816	0.4752	0.4580	0.4828
Overall	0.8205	0.7818	0.7731	0.7342	0.7774
SWC-ICP $k = 50\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.9639	0.9588	0.8941	0.9542
$\sigma = 0.01$	0.9445	0.9033	0.8784	0.8017	0.8820
$\sigma = 0.05$	0.5142	0.4795	0.4691	0.4362	0.4747
Overall	0.8196	0.7823	0.7687	0.7107	0.7703
SWC-ICP $k = 15\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.9613	0.9460	0.8422	0.9374
$\sigma = 0.01$	0.9445	0.8749	0.8444	0.7328	0.8492
$\sigma = 0.05$	0.5008	0.4428	0.4188	0.3596	0.4305
Overall	0.8151	0.7597	0.7364	0.6449	0.7390
SWC-ICP $k = 5\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.9647	0.9100	0.7835	0.9146
$\sigma = 0.01$	0.9445	0.8563	0.7744	0.6282	0.8008
$\sigma = 0.05$	0.4512	0.3214	0.2851	0.2387	0.3241
Overall	0.7986	0.7141	0.6565	0.5502	0.6798
SWC-ICP $k = 1\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.9867	0.9833	0.8373	0.9518
$\sigma = 0.01$	0.5729	0.2736	0.1806	0.1091	0.2840
$\sigma = 0.05$	0.0860	0.0417	0.0414	0.0425	0.0529
Overall	0.5530	0.4340	0.4018	0.3299	0.4297
GMM	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	0.5370	0.5385	0.5210	0.5187	0.5288
$\sigma = 0.01$	0.5166	0.4971	0.4918	0.4999	0.5013
$\sigma = 0.05$	0.2803	0.2725	0.2799	0.2744	0.2768
Overall	0.4446	0.4360	0.4309	0.4310	0.4356
S4PCS + GMM	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.4139	0.3953	0.4268	0.5590
$\sigma = 0.01$	0.9450	0.3890	0.3657	0.3802	0.5200
$\sigma = 0.05$	0.4493	0.2009	0.1799	0.1835	0.2534
Overall	0.7981	0.3346	0.3136	0.3301	0.4441
BESL	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	0.3446	0.2673	0.2302	0.2119	0.2635
$\sigma = 0.01$	0.3049	0.2145	0.2303	0.2091	0.2397
$\sigma = 0.05$	0.1642	0.1400	0.1355	0.1188	0.1396
Overall	0.2712	0.2073	0.1987	0.1799	0.2143
ICP-CTSF F. $k = 75\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	1.0000	1.0000	0.9473	0.9868
$\sigma = 0.01$	0.9451	0.9372	0.9348	0.8822	0.9248
$\sigma = 0.05$	0.5229	0.5044	0.5024	0.4678	0.4994
Overall	0.8227	0.8139	0.8124	0.7658	0.8037
Sparse ICP	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	0.4610	0.3899	0.3508	0.3030	0.3762
$\sigma = 0.01$	0.4368	0.3558	0.3551	0.3002	0.3620
$\sigma = 0.05$	0.2151	0.1933	0.1800	0.1481	0.1841
Overall	0.3710	0.3130	0.2953	0.2504	0.3074
Sparse ICP CTSF	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.7222	0.7083	0.6556	0.7715
$\sigma = 0.01$	0.8817	0.6536	0.6475	0.5973	0.6950
$\sigma = 0.05$	0.3840	0.3387	0.3264	0.3064	0.3388
Overall	0.7552	0.5715	0.5607	0.5198	0.6018

Table C.1: Score by noise and outlier level from different methods on the Bunny. The Overall row is the average for a fixed amount of outliers, while the Overall column is the average for a fixed noise value.

SWC-ICP $k = 100\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.7125	0.8386	0.6899	0.8103
$\sigma = 0.01$	0.6979	0.5590	0.6099	0.4770	0.5860
$\sigma = 0.05$	0.4555	0.4320	0.4116	0.3161	0.4038
Overall	0.7178	0.5678	0.6200	0.4943	0.6000
SWC-ICP $k = 75\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.7487	0.8673	0.9267	0.8857
$\sigma = 0.01$	0.6979	0.5442	0.6088	0.6243	0.6188
$\sigma = 0.05$	0.4545	0.4365	0.4408	0.4263	0.4395
Overall	0.7175	0.5765	0.6390	0.6591	0.6480
SWC-ICP $k = 50\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.7492	0.8715	0.9408	0.8905
$\sigma = 0.01$	0.6980	0.5502	0.6143	0.6272	0.6224
$\sigma = 0.05$	0.4534	0.4377	0.4380	0.4237	0.4382
Overall	0.7171	0.5789	0.6413	0.6639	0.6503
SWC-ICP $k = 15\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.7509	0.8858	0.8403	0.8692
$\sigma = 0.01$	0.6980	0.5317	0.5986	0.5294	0.5894
$\sigma = 0.05$	0.4482	0.3927	0.3468	0.3200	0.3770
Overall	0.7154	0.5584	0.6104	0.5632	0.6119
SWC-ICP $k = 5\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.7452	0.8691	0.6472	0.8154
$\sigma = 0.01$	0.6979	0.5414	0.5494	0.3621	0.5377
$\sigma = 0.05$	0.4272	0.3254	0.2913	0.2141	0.3145
Overall	0.7084	0.5373	0.5699	0.4078	0.5559
SWC-ICP $k = 1\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.7553	0.8697	0.9556	0.8952
$\sigma = 0.01$	0.5050	0.2710	0.2912	0.2151	0.3206
$\sigma = 0.05$	0.0997	0.0902	0.0831	0.0879	0.0903
Overall	0.5349	0.3722	0.4147	0.4196	0.4353
GMM	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	0.3793	0.3673	0.3734	0.3862	0.3765
$\sigma = 0.01$	0.2952	0.2769	0.2746	0.2754	0.2805
$\sigma = 0.05$	0.2086	0.2149	0.2076	0.2099	0.2103
Overall	0.2943	0.2864	0.2852	0.2905	0.2891
S4PCS + GMM	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	0.9415	0.4504	0.4689	0.5515	0.6031
$\sigma = 0.01$	0.6759	0.3312	0.3619	0.4024	0.4429
$\sigma = 0.05$	0.4488	0.2278	0.2565	0.2623	0.2989
Overall	0.6887	0.3365	0.3624	0.4054	0.4483
BESL	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	0.2817	0.2504	0.2463	0.2505	0.2572
$\sigma = 0.01$	0.2212	0.2040	0.2067	0.2032	0.2087
$\sigma = 0.05$	0.1722	0.1702	0.1713	0.1734	0.1718
Overall	0.2250	0.2082	0.2081	0.2090	0.2126
ICP-CTSF F. $k = 50\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.9963	0.8937	0.8241	0.9285
$\sigma = 0.01$	0.6982	0.6643	0.5932	0.5369	0.6231
$\sigma = 0.05$	0.4575	0.4360	0.3791	0.3599	0.4082
Overall	0.7186	0.6989	0.6220	0.5736	0.6533
Sparse ICP	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	0.3564	0.3454	0.3229	0.3013	0.3315
$\sigma = 0.01$	0.2710	0.2604	0.2537	0.2368	0.2555
$\sigma = 0.05$	0.1907	0.1942	0.1906	0.1900	0.1914
Overall	0.2727	0.2667	0.2558	0.2427	0.2595
Sparse ICP CTSF F.	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.8583	0.7646	0.6649	0.8219
$\sigma = 0.01$	0.6965	0.5868	0.5200	0.4605	0.5660
$\sigma = 0.05$	0.4098	0.3428	0.3143	0.2674	0.3358
Overall	0.7021	0.5960	0.5329	0.4734	0.5773

Table C.2: Score by noise and outlier level from different methods on the Octopus. The Overall row is the average for a fixed amount of outliers, while the Overall column is the average for a fixed noise value.

SWC-ICP $k = 100\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.5631	0.4871	0.4089	0.6148
$\sigma = 0.01$	0.7344	0.4972	0.4356	0.3837	0.5127
$\sigma = 0.05$	0.4281	0.3574	0.3356	0.3201	0.3603
Overall	0.7208	0.4726	0.4195	0.3709	0.4959
SWC-ICP $k = 75\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.6862	0.5958	0.4805	0.6906
$\sigma = 0.01$	0.7597	0.5303	0.4920	0.4538	0.5589
$\sigma = 0.05$	0.4500	0.3835	0.3630	0.3532	0.3875
Overall	0.7366	0.5334	0.4836	0.4292	0.5457
SWC-ICP $k = 50\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.7426	0.6269	0.5035	0.7182
$\sigma = 0.01$	0.7459	0.5587	0.5184	0.4582	0.5703
$\sigma = 0.05$	0.4527	0.4152	0.3996	0.3793	0.4117
Overall	0.7329	0.5722	0.5150	0.4470	0.5667
SWC-ICP $k = 15\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.6133	0.5116	0.4254	0.6376
$\sigma = 0.01$	0.6003	0.4232	0.3950	0.3757	0.4486
$\sigma = 0.05$	0.3860	0.3359	0.3328	0.3119	0.3417
Overall	0.6621	0.4575	0.4131	0.3710	0.4759
SWC-ICP $k = 5\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.6469	0.5080	0.3870	0.6355
$\sigma = 0.01$	0.5010	0.3656	0.3447	0.3333	0.3861
$\sigma = 0.05$	0.3136	0.2844	0.2884	0.2723	0.2897
Overall	0.6049	0.4323	0.3803	0.3309	0.4371
SWC-ICP $k = 1\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.9356	0.7874	0.4521	0.7938
$\sigma = 0.01$	0.1572	0.1129	0.1056	0.0891	0.1162
$\sigma = 0.05$	0.0866	0.0825	0.0814	0.0840	0.0836
Overall	0.4146	0.3770	0.3248	0.2084	0.3312
GMM	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	0.5251	0.5252	0.5113	0.5221	0.5209
$\sigma = 0.01$	0.4333	0.4303	0.4375	0.4284	0.4324
$\sigma = 0.05$	0.2574	0.2721	0.2688	0.2692	0.2669
Overall	0.4053	0.4092	0.4059	0.4066	0.4067
S4PCS + GMM	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	0.9977	0.6142	0.5772	0.6468	0.7090
$\sigma = 0.01$	0.8115	0.4565	0.4939	0.5226	0.5717
$\sigma = 0.05$	0.4362	0.2944	0.2814	0.2899	0.3255
Overall	0.7485	0.4550	0.4509	0.4861	0.5353
BESL	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	0.4752	0.3931	0.4194	0.4340	0.4304
$\sigma = 0.01$	0.3819	0.3328	0.3492	0.3832	0.3618
$\sigma = 0.05$	0.2316	0.2268	0.2304	0.2497	0.2346
Overall	0.3629	0.3176	0.3330	0.3556	0.3423
ICP-CTSF F. $k = 50\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	1.0000	0.9973	0.8562	0.9634
$\sigma = 0.01$	0.8134	0.7797	0.7751	0.6353	0.7509
$\sigma = 0.05$	0.4700	0.4552	0.4439	0.3858	0.4387
Overall	0.7611	0.7450	0.7388	0.6258	0.7177
Sparse ICP	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	0.5385	0.4994	0.4725	0.4460	0.4891
$\sigma = 0.01$	0.4561	0.4364	0.4149	0.3797	0.4218
$\sigma = 0.05$	0.2535	0.2640	0.2505	0.2473	0.2538
Overall	0.4160	0.3999	0.3793	0.3576	0.3882
Sparse ICP CTSF F.	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.7452	0.7106	0.6665	0.7806
$\sigma = 0.01$	0.6439	0.5811	0.5637	0.5517	0.5851
$\sigma = 0.05$	0.3361	0.3178	0.3140	0.2906	0.3146
Overall	0.6600	0.5481	0.5295	0.5029	0.5601

Table C.3: Score by noise and outlier level from different methods on the Happy. The Overall row is the average for a fixed amount of outliers, while the Overall column is the average for a fixed noise value.

SWC-ICP $k = 100\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.8126	0.6068	0.4650	0.7211
$\sigma = 0.01$	0.8364	0.5532	0.4669	0.3840	0.5601
$\sigma = 0.05$	0.3557	0.2887	0.2684	0.2301	0.2857
Overall	0.7307	0.5532	0.4474	0.3597	0.5227
SWC-ICP $k = 75\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.9779	0.9410	0.5916	0.8776
$\sigma = 0.01$	0.8363	0.7336	0.6202	0.4413	0.6579
$\sigma = 0.05$	0.4222	0.3255	0.3019	0.2312	0.3202
Overall	0.7528	0.6806	0.6210	0.4214	0.6189
SWC-ICP $k = 50\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.9856	0.9717	0.6961	0.9134
$\sigma = 0.01$	0.8364	0.7530	0.6639	0.5028	0.6890
$\sigma = 0.05$	0.4581	0.3469	0.3344	0.3158	0.3639
Overall	0.7648	0.6968	0.6567	0.5049	0.6557
SWC-ICP $k = 15\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.9772	0.9208	0.6209	0.8797
$\sigma = 0.01$	0.8363	0.6118	0.5166	0.4650	0.6074
$\sigma = 0.05$	0.3420	0.2479	0.2702	0.2788	0.2848
Overall	0.7261	0.6137	0.5692	0.4549	0.5909
SWC-ICP $k = 5\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.9809	0.9537	0.6073	0.8855
$\sigma = 0.01$	0.6984	0.4790	0.4634	0.4106	0.5128
$\sigma = 0.05$	0.2810	0.2413	0.2315	0.2032	0.2392
Overall	0.6598	0.5677	0.5495	0.4071	0.5460
SWC-ICP $k = 1\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	0.9784	0.9971	0.9486	0.9810
$\sigma = 0.01$	0.3568	0.1792	0.1704	0.1816	0.2220
$\sigma = 0.05$	0.1314	0.1052	0.1006	0.1092	0.1116
Overall	0.4961	0.4230	0.4227	0.4131	0.4387
GMM	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	0.5283	0.5438	0.5350	0.5360	0.5358
$\sigma = 0.01$	0.4606	0.4563	0.4472	0.4420	0.4515
$\sigma = 0.05$	0.2680	0.2713	0.2743	0.2656	0.2698
Overall	0.4190	0.4238	0.4188	0.4145	0.4190
S4PCS + GMM	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	0.9790	0.4568	0.4749	0.4636	0.5936
$\sigma = 0.01$	0.7766	0.3497	0.3680	0.3907	0.4713
$\sigma = 0.05$	0.2903	0.2054	0.2358	0.2106	0.2356
Overall	0.6820	0.3373	0.3596	0.3550	0.4335
BESL	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	0.4124	0.3664	0.3922	0.3940	0.3912
$\sigma = 0.01$	0.3676	0.3150	0.3268	0.3214	0.3327
$\sigma = 0.05$	0.2237	0.2074	0.2176	0.2177	0.2166
Overall	0.3345	0.2963	0.3122	0.3114	0.3136
S4PCS	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	0.9835	0.4275	0.4349	0.4317	0.5694
$\sigma = 0.01$	0.7663	0.3679	0.3301	0.3821	0.4616
$\sigma = 0.05$	0.3039	0.1977	0.2024	0.2087	0.2282
Overall	0.6846	0.3310	0.3225	0.3410	0.4198
ICP-CTSF F. $k = 50\%$	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	1.0000	1.0000	0.9567	0.6491	0.9015
$\sigma = 0.01$	0.8362	0.7556	0.7032	0.5321	0.7068
$\sigma = 0.05$	0.4627	0.4063	0.3546	0.2925	0.3789
Overall	0.7663	0.7215	0.6715	0.4912	0.6626
Sparse ICP	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	0.5359	0.4717	0.4609	0.3840	0.4631
$\sigma = 0.01$	0.4637	0.4003	0.3810	0.3396	0.3962
$\sigma = 0.05$	0.2681	0.2528	0.2448	0.2253	0.2478
Overall	0.4226	0.3749	0.3622	0.3165	0.3691
Sparse ICP CTSF F.	0.00	10.00	20.00	50.00	**Overall**
$\sigma = 0.00$	0.8904	0.6207	0.5739	0.5712	0.6640
$\sigma = 0.01$	0.5525	0.4938	0.5034	0.4970	0.5117
$\sigma = 0.05$	0.2797	0.2799	0.2858	0.2753	0.2802
Overall	0.5742	0.4648	0.4544	0.4479	0.4853

Table C.4: Score by noise and outlier level from different methods on the Genus. The Overall row is the average for a fixed amount of outliers, while the Overall column is the average for a fixed noise value.