

International Journal of Image and Graphics
© World Scientific Publishing Company

Wide Angle Rigid Registration using a Comparative Tensor Shape Factor

Luciano W. X. Cejnog

*Post-Graduation Program on Computer Science, Universidade Federal de Juiz de Fora
36036-900, Juiz de Fora, Minas Gerais, Brazil
lxavier@ice.ufjf.br*

Fernando A. A. Yamada

*Post-Graduation Program on Computer Science, Universidade Federal de Juiz de Fora
36036-900, Juiz de Fora, Minas Gerais, Brazil
akio@ice.ufjf.br*

Marcelo B. Vieira

*Post-Graduation Program on Computer Science, Universidade Federal de Juiz de Fora
36036-900, Juiz de Fora, Minas Gerais, Brazil
marcelo.bernardes@ufjf.edu.br*

Received 26 April 2016

Revised (Day Month Year)

Accepted 23 November 2016

This work aims to enhance a classic method for rigid registration, the Iterative Closest Point (ICP), modifying the closest point search in order to consider approximated information of local geometry combined to the Euclidean distance, originally used. For this, a preprocessing stage is applied, in which the local geometry is encoded in second-order orientation tensors. We define the CTSE, a similarity factor between tensors. Our method uses a strategy of weight variation between the CTSE and the Euclidean distance, in order to establish correspondences. Quantitative tests were made in point clouds with different geometric features, with variable levels of additive noise and outliers and in partial overlapping situations. Results show that the proposed modification increases the convergence probability of the method for higher angles, making the method comparable to state-of-art techniques.

Keywords: Rigid registration; iterative closest point; orientation tensor; shape dissimilarity; coarse-to-fine alignment.

1. Introduction

Surface registration is the alignment of 3D surfaces into a common coordinate system. It is an important process on computer geometry applications, inserted into the processing pipeline of geometry acquisition for robot vision and many medical image applications. Rigid registration is a common sub-problem, constraining the solution into a rotation and translation that best align two surfaces, consisting of

a six degrees of freedom problem.

The first successful solution for the rigid registration problem was Besl and McKay's Iterative Closest Point (ICP) ¹. This method takes as input two point sets, named M and D , and is executed in two steps: matching and transformation estimation. In the matching step, each point of M is associated to its closest point on D , generating a list of correspondences. The transformation estimation uses the covariance between M and the correspondences set to estimate the transformation which minimizes the L_2 residual error between M and D . These two steps are repeated until a local optima or a minimal error threshold is reached.

The original ICP method is able to retrieve optimal alignment when both sets represent completely the same object and there is a small rotation and translation between them. In practice, when used in surface reconstruction applications, the point sets are mostly partially overlapped and subject to the presence of outliers and additive noise on the scanner raw data. Notably, these are non-optimal scenarios in terms of minimizing the L_2 error, and the original ICP tends to perform poorly. The ICP algorithm also assumes that the angular displacement between *Model* and *Data* point sets is small, since it is designed for fine alignment. If this assumption is not accomplished, the method is not guaranteed to yield good results.

In this work, we use a 3D mathematical morphology method based on the work of Vieira ² to estimate a second-order tensor which describes the local covariance of point dispositions on a mesh. We suggest some optimizations to the tensor estimation process in order to achieve higher precision on the normals and consequently on the quality of the estimated tensor. We define a Comparative Tensor Shape Factor (CTSF) capable of representing the shape similarity between two second-order tensors, i.e. between two distinct local planar continuity inferences. The CTSF is invariant to rigid transformations making it suitable for rigid registration. We modify the matching step of the ICP in order to use a linear combination of the CTSF and the euclidean distance to achieve more precise correspondences on the closest point search. We also propose an heuristic optimization of the ICP, varying the relative weight between the CTSF and the euclidean distance when a local optima is reached. This works like a coarse-to-fine strategy and enhances the convergence rate of the alignment between two point sets. Our method is evaluated in situations of large angle displacements, additive noise, outliers and with partial overlapping sets, comparing the results obtained to the original ICP and the recent approach proposed by Bouaziz *et al.* ³. We statistically show that our method is capable to reach convergence for wide angles and tolerates a reasonable amount of outliers and additive noise.

The main contributions of this work are: 1) the CTSF to compare the shape of two tensors representing local geometry, and 2) the strategy to vary the relative weight of the CTSF and the euclidean distance, to guide the computation of successive rigid transformations that improves the chance of global convergence. Our proposals greatly enhance the registration of point clouds mismatched by wide angle rotations. The secondary but important contributions include: a method for

finding smooth normals given a k -neighborhood, a clearly stated comparison protocol to evaluate registration methods, a point cloud database composed by a large amount of events that can be used for future quantitative comparisons with our method.

1.1. Related Works

Since the introduction of the ICP, many approaches have been proposed addressing issues for the rigid registration problem. Due to the large variety of applications, the notation and the evaluation procedures used are not standardized, which highlights the importance of surveys and benchmarks.

Among these surveys, Rusinkiewicz⁴ addresses variations on six possible stages of the ICP. Following their taxonomy, our method modifies the *matching* stage. Salvi⁵ proposes a taxonomy for rigid registration methods, classifying them into coarse and fine. The recent work of Tam⁶ formulates the registration problem as data fitting, presenting methods for rigid and non-rigid registration.

In this work, we will follow the taxonomy of Salvi⁵, and summarize the state-of-art methods on coarse and fine registration separately. While coarse registration aims to compute an initial estimation of the rigid transformation, fine registration assumes that the point clouds are pre-aligned and seek a more accurate solution.

Coarse methods are usually based on the matching of reliable correspondences. Thus, this problem is closely related to keypoint detection and description on 3D point clouds, and good keypoint approaches^{7 8 9 10} tend to yield good results on coarse registration. The literature presents benchmarks for keypoint detection¹¹ that can be used for coarse registration. An alternative to keypoints is the use of local shape descriptors, such as Spin Images¹² and the Hybrid Dimensionality-Reduction Shape Descriptor¹³, that performs an angle-preserving parametrization in order to represent each point with a 2D Harmonic Map. Usually, coarse methods use different search strategies to find the transformation, like RANSAC-based methods¹⁴. Among them, the most successful solution is the 4PCS method¹⁵, based on finding small congruent 4-point subsets, which achieves good results even in the presence of a high amount of outliers and additive noise. Although originally designed for coarse registration, 4PCS reaches results comparable to fine registration methods. More recently, a method named Super-4PCS¹⁶ was proposed. This method enhances the 4PCS in terms of speed, achieving linear complexity by the use of an efficient data indexing, and keeping the alignment quality even in cases of low overlap. More details on coarse registration methods can be found in the recent survey of Diez¹⁷, which proposes a pipelined classification and presents the current state-of-art coarse methods.

On fine registration methods, the general effort of the literature is in optimizations of the ICP method, aiming to enhance its performance in terms of speed, robustness and accuracy. The survey of Rusinkiewicz and Levoy⁴ identifies six possible stages of the algorithm in which optimizations can be made: selecting a

subset of points^{4, 18, 19}, changing the closest point criterion^{20, 21, 22}, weighting the correspondences^{4, 20}, rejecting pairs^{23, 24}, assigning an error metric^{3, 25, 26, 27} and minimizing this error metric^{3, 26, 28, 29}. Taking into account their taxonomy, our method only modifies the closest point criterion by combining the CTSF and the euclidean distance.

In this aspect, we propose a novel successive approximation strategy that dynamically changes the closest point criterion, varying the relative weight of the shape factor and the euclidean distance. The weights are altered every time the algorithm reaches a local optimum point, increasing the euclidean distance influence. This strategy guides the algorithm from a coarse alignment, based on tensor similarity, to a fine alignment, computed from the euclidean distance.

Many works propose modifications on the closest point criterion, using other information, such as color^{20, 21} or geometric compatibility²², to enhance the quality of the correspondences. Godin²⁰ proposes a method called Iterative Compatible Closest Point that alters the distance metric to consider also similarity by color, also extensible to other intensity invariants. Sharp *et al.*'s ICP-IF²² introduces the use of geometric invariant features such as curvature, moment invariants and spherical harmonics invariants on the ICP, altering the closest point criterion to a linear combination of euclidean distance and feature compatibility. The features are weighted by a factor α^2 , whose value is set at each iteration as the mean squared euclidean distance from the data points to their closest points. This scheme implicitly reduces the weight of the invariant feature factor as the transformation bring both point clouds closer to each other. However, this work has as limitation the need of a point grid to compute the invariant features, which makes it only applicable on range image alignment. Our work uses a highly similar approach, considering a linear combination between the factors, but we use a different strategy of variation of the weight. Also, the CTSF is computed from a k -neighborhood that can be obtained from raw point data, which extends the application of our method to point clouds in general.

Discarding bad correspondences is a common strategy to deal with outliers and missing data. One of the methods that use it is the Trimmed ICP²³, which consists on discarding the $\epsilon\%$ correspondences that have worse euclidean distance values. An alternative is to discard correspondences according to a distance threshold instead of a percentage of the mesh. Due to its simplicity, the Trimmed ICP is used as a standard strategy to deal with partial overlapping, and some works adapt it to discard correspondences with different distance metrics^{27, 24}.

The choice of the error metric and its minimization plays an important role on the quality of the result. The literature contains approaches that explore other minimization methods and other error metrics than the L_2 norm. The RICP²⁵ proposes the use of a Least Median Squares regression to enhance the performance in the presence of outliers and with missing data. The LM-ICP²⁶ uses the traditional Levenberg-Marquardt non-linear optimization algorithm on the parameter space,

allowing to use robust kernels and other error metrics. The results achieved are comparable to the original method in precision and speed. The EM-ICP³⁰ models the point clouds as Gaussian Mixture Models, adapting the ICP to an Expectation-Maximization process.

Among the recent approaches, the Sparse ICP³ uses L_p norms to change the error metric. The sparsity induced by the use of L_p norms when $p \in [0, 1]$ reduces the influence of outliers on the transformation estimative, but the optimization problem becomes non-convex and the minimization is made by Augmented Lagrangian methods. The Generalized ICP²⁸ associates each point to a covariance matrix, computed from a principal component analysis on its nearest neighbors, and assumes that the error between each point and its correspondence is drawn from a Gaussian distribution. The transformation is then estimated using a maximum likelihood estimation process. This method was later expanded³¹ to build a Multi-Channel kernel descriptor capable of incorporating features of the images, such as reflectance coefficients and color, to enhance the quality of the correspondences. The Anisotropic ICP²⁷ computes a cross-covariance matrix for each point and uses this information on the closest point computation and on the error minimization, adapting the problem to deal with anisotropic and inhomogeneous errors.

Our work links to the Generalized ICP and Anisotropic ICP since these works also express local geometry information of points on covariance matrices. However, to our knowledge, our method is the first rigid registration approach to treat local covariance matrices as second-order tensors and use invariant tensor properties to compute similarity between patches on both sets. There is a previous method³², different than the ICP, that uses second-order tensors computed by a voting process to solve the rigid registration problem, but this work differs from ours on the use of a complex geometric algebra formulation, while we use simple tensor coefficients computed directly from its eigenvalues.

Alternative solutions have been explored in some recent approaches, with competitive results. The GoICP²⁹ achieves globally optimal solutions under the L_2 error for any initial position, integrating the ICP minimization with a branch-and-bound search on a 3D motion space. The Trimmed ICP was later extended to a Lie Group parametrization formulation, the LieTrICP²⁴, which improves the accuracy on partial overlapping situations and select automatically the amount of points discarded, reformulating the error function to deal with anisotropic errors through the use of Lie algebra properties. The LieTrICP is able to retrieve anisotropic scale transformations, besides the rigid transformation. Another recent approach³³ uses a novel game-theoretical framework on the establishment of robust correspondences, introducing the Integral Hash coarse descriptor and reaching competitive results either for coarse and fine registration, in situations with different levels of occlusion and noise.

In fine rigid registration literature, in order to highlight the good performance of the proposed method, most works present only qualitative comparisons, usually made with a small number of point clouds and on situations predetermined by

the author. This type of evaluation is generally inconclusive, because the behavior observed in a single case may not describe the overall performance of the method. Our work, in contrast, presents an experimental setup for quantitative evaluation. For this, we generate a large dataset composed of transformations between two point clouds that include cases with wide angle initialization, additive noise, outliers and partial overlapping. With this dataset and a well stated error measurement protocol, we can characterize the behavior of the methods with different parameter values and on different situations.

2. Proposed Method

To estimate the local geometry of both point sets, a preprocessing stage based on a tensor voting process is applied, aiming to estimate orientation tensors that represent the local geometry. The basic assumption is that when two points belonging to different viewpoints of the same object have the same neighborhood disposition, they are likely to represent a common region of one object in different views. Thus, the estimation of those orientation tensors generates additional information, that can be used on the ICP algorithm.

This preprocessing method has as input the single 3D position of each point, not requiring any other information. Since the method relies on nearest neighbors to compute the tensors, we assume that the input point cloud represents a surface. Volumetric representations would add internal points among the neighbor lists, which would affect directly the output of the tensor estimation.

2.1. Tensor Voting Framework

The preprocessing stage used in this work computes a second-order tensor for each point, representing its local geometric disposition. It is based on the method proposed by Vieira et al.², and uses a tensor voting framework based on the well known work of Medioni et. al.³⁴. Tensor voting has many formulations and deals efficiently with a wide number of problems on computer vision and computer graphics, including surface reconstruction, stereo vision, edge detection, data repairing and optical flow, and is acknowledged for its robustness to outliers.

The tensor voting algorithm is the accumulation of the influence of the local neighborhood on a point, which generates a tensor that encodes an estimation of the local distribution of a feature. For each point, a voting field is generated. The point propagates its information as a vote to all his neighbors, and uses the information cast by them in order to generate the tensor.

The general formulation of the tensor voting algorithm is described by:

$$\mathbf{T}_p = \sum_{q \in N(p)} f_{qp} \cdot \mathbf{T}_{qp},$$

where $N(p)$ is the neighborhood of the point, f_{qp} is a scalar field that weights the influence of each neighbor $q \in N(p)$, and \mathbf{T}_{qp} is the vote that each neighbor q casts

on the point p . In our case, the neighborhood of each point is represented by the list $L_k(p)$ of its k nearest-neighbors sorted by their euclidean distances. Conversely, we represent as $L_k^{-1}(p)$ the list of points that contains p into its k nearest neighbors. The method assumes that these lists are previously computed.

Our implementation of the tensor voting is composed by two steps. In the first step, a coarse estimation of the main curvature directions is obtained for each point. In the second, the previous estimation is enhanced, increasing the influence of coplanar structures on the neighborhood, and consequently the local planarity of the tensor.

2.1.1. First pass - Radial structuring element

The first step builds for each point p a second-order tensor \mathbf{T}_p , which accumulates the weighted sum of stick tensors built from the vectors \vec{pq} , for each neighbor $q \in L_k(p)$. The influence function f_{qp} is a Gaussian decay proportional to the euclidean distance between p and q with standard deviation σ_p . This deviation is proportional to the closeness of the neighbors, such that the farthest neighbor q_f has a weak influence ϵ , which in our case is set as $\epsilon = 0.01$.

$$\sigma_p = \sqrt{\frac{\|\vec{pq}_f\|^2}{\ln \epsilon}}. \quad (1)$$

The tensors \mathbf{T}_{qp} are stick tensors built from a tensor product of a normalized vector \hat{v}_{qp} with itself transposed. In this step, the vote direction \hat{v}_{qp} is the normalized vector \vec{pq} .

The output of the first step is the set of tensors \mathbf{T}_p :

$$\mathbf{T}_p = \sum_{q \in L_k(p)} f_{qp} \cdot \hat{v}_{qp} \cdot \hat{v}_{qp}^T = \sum_{q \in L_k(p)} e^{-\frac{\|\vec{pq}\|^2}{\sigma_p^2}} \cdot \vec{pq} \cdot \vec{pq}^T, \quad (2)$$

where the Gaussian influence function f_{qp} is proportional to the euclidean distance between p and q , with standard deviation σ_p , as in Eq. (1). The Eq. (2) corresponds to the application of a 3D isotropic radial structuring element. The third main direction of \mathbf{T}_p is a rough estimation of the normal on points with planar neighborhood.

2.1.2. Second pass - Coplanar structuring element

The tensors \mathbf{T}_p obtained in the first step are used as input to the second step. Here, another structuring element is applied on the point set, in order to enforce the influence of local coplanar structures. Differently from the first step, here each point p casts a stick vote on its neighbors q , based on a vector \hat{v}_{pq} , and its influence on the tensor \mathbf{S}_q is given by a function f_{pq} , proportional to the coplanarity between p and q . This function ensures that points aligned to the tangent plane have higher influence.

8 *Cejnog et. al.*

To estimate \hat{v}_{pq} and f_{pq} , we first bring all points to a different coordinate system, where the axes $(\hat{x}, \hat{y}, \hat{z})$ are respectively aligned with the normalized eigenvectors $(\hat{e}_1, \hat{e}_2, \hat{e}_3)$ of \mathbf{T}_p . The transformation of each neighbor q to this system is obtained by the application of a rotation matrix R_p built from \hat{e}_1 , \hat{e}_2 and \hat{e}_3 on the vector \vec{pq} , generating a new point denoted q' . In this system, the correspondent point p' of the point p is the origin. Note that \vec{pq} is a column vector (3×1) .

$$q' = \begin{bmatrix} \hat{e}_1 \\ \hat{e}_2 \\ \hat{e}_3 \end{bmatrix} \cdot (\vec{pq}).$$

The next step is to express q' in spherical coordinates:

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

For each neighbor q' , there is a unique ellipsoid E with eccentricity $\tan \alpha_{ellip}$ that is centered over the \hat{z}' -axis and tangent to p' and q' . Figure 1 shows ellipses with different α_{ellip} parameters passing through an arbitrary point, in a $2D$ representation.

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:

$$g = \tan \alpha_{ellip}$$

$$d_e(p, q) = \rho_{q'} \cos \phi_{q'} \left(1 + \left(2 - \frac{1}{g^2} \right) \tan^2 \phi_{q'} \right)^{\frac{g^2}{2g^2-1}}. \quad (3)$$

To avoid numerical instabilities on the computation of $d_e(p, q)$, the angle α_{ellip} must be at least $\tan^{-1} \frac{\sqrt{2}}{2} \approx 35.26^\circ$. Smaller values of α_{ellip} would result in a negative base and a negative exponent, yielding an invalid operation.

From the spherical distance defined in Eq. (3) and the standard deviation σ_p from Eq. (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 ϵ .

Points with $\tan \phi_{q'} > \tan \phi_{max}$ are discarded to reduce the influence of points misaligned to the tangent plane.

$$f_{pq} = \begin{cases} \frac{-d_e(p, q)}{\sigma_p^2}, & \phi_{q'} \leq \phi_{max}, \\ 0.0, & \phi_{q'} > \phi_{max}. \end{cases}$$

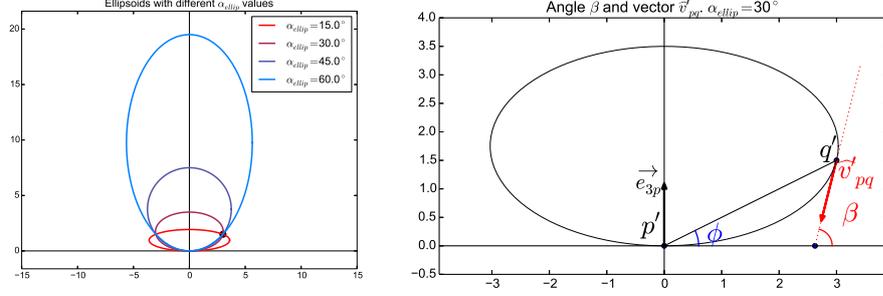


Fig. 1: Left: Family of ellipses (2D representation) passing through the point $(3.0, 1.5)$ with different values of α_{ellip} . Right: 2D geometric representation of the angles ϕ , β and vector \widehat{v}_{pq} of an arbitrary point q' . Note that \widehat{v}_{pq} is normalized.

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 E at the point q' . The direction defined by the angle $\beta_{q'}$ is assumed to be an estimation of a vector that belongs to the tangent plane of q' . In function of α_{ellip} and $\phi_{q'}$, $\beta_{q'}$ is:

$$\beta_{q'} = \tan^{-1} \frac{2 \cdot \tan^2 \alpha_{ellip} \cdot \tan \phi_{q'}}{\tan^2 \alpha_{ellip} - \tan^2 \phi_{q'}}.$$

By the replacement of the angle $\phi_{q'}$ for $\beta_{q'}$ and the conversion of q' back to euclidean coordinates, we obtain a vector $\widehat{v}'_{pq} = (\cos \theta_{q'} \cdot \cos \beta_{q'})\widehat{i} + (\sin \theta_{q'} \cdot \cos \beta_{q'})\widehat{j} + (\sin \beta_{q'})\widehat{k}$. The vote vector \widehat{v}_{pq} is obtained by the application of the inverse rotation matrix R_p^{-1} .

Figure 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 $\alpha_{ellip} = 30^\circ$.

The vote \mathbf{S}_{pq} cast by p on a neighbor q is expressed by:

$$\mathbf{S}_{pq} = f_{pq} \cdot \widehat{v}_{pq} \cdot \widehat{v}_{pq}^T.$$

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:

$$\mathbf{S}_p = \sum_{q \in L_k^{-1}(p)} \mathbf{S}_{qp} = \sum_{q \in L_k^{-1}(p)} f_{qp} \cdot \widehat{v}_{qp} \cdot \widehat{v}_{qp}^T.$$

2.1.3. Improvements and discussion about the parameters

The differences between the original method ² and the one proposed and used in this work are:

- (1) We propose an iterative reapplication of the second step, through the recalculation of the tensors using with \mathbf{S}_p as input. With this process, the planarity

properties of the final tensors are continuously enhanced. The method stops when the average c_p stops improving or a maximum number of iterations is reached. In all experiments, we use a maximum of 100 iterations.

- (2) The original method uses all the points of the cloud as neighborhood and sets a default value for the standard deviation, equal for all points. We use a k -neighborhood and set the value of σ proportional to the distance to the k -th neighbor
- (3) The original method uses the normal information for surface reconstruction. Thus, the computation of the vector \hat{v}_{pq} is set to encode the normal in the first main direction of the tensor. Our implementation, in turn, encodes the normal in the third eigenvector, since the region geometry is as important as the normal itself in our work.
- (4) The original approach treats α_{ellip} and ϕ_{max} as the same variable, while we choose to deattach them.

Figure 2b shows that the method is effective in acquiring better geometry information of planar regions even with unrealistic outlier amounts, which is highlighted on the head of the Octopus. Additive noise is naturally a hard scenario for the tensor estimation, since the smoothness of the neighborhood is compromised. However, with small amounts of additive noise, the method is still capable of encoding planarity information. In Figure 2c, tensors on the head of the octopus still have large planarity coefficient values.

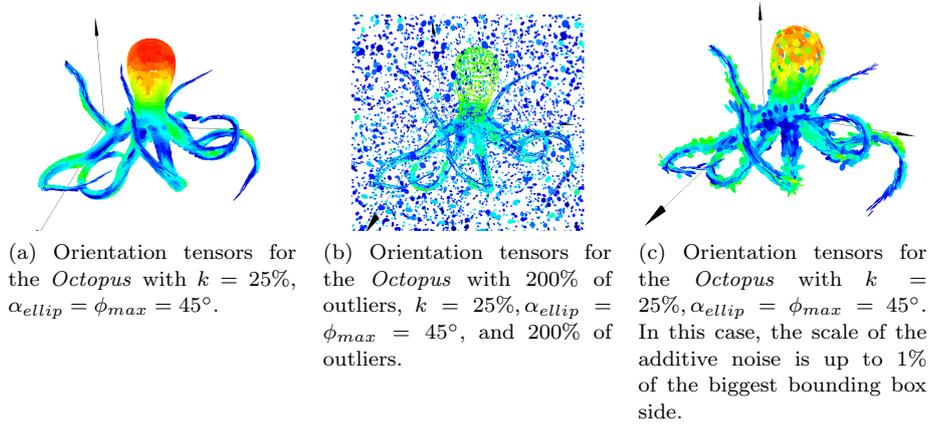


Fig. 2: Different examples for tensor estimation.

The number of neighbors k is an important parameter on the tensor estimation process. This value affects the scale of the neighborhood and the computation of σ_p . Figure 3 shows examples of output tensors of the method for different values

of k on the Bunny point cloud. If the value of k is very small, the tensor may have insufficient information of the local geometry, as shown in the result with $k = 1\%$. In counterpart, for $k = 100\%$, the resulting tensor encodes information on distant points of the set. This explains the fact that the planarity of the tensors is generally lower with the increase of k . Larger neighborhoods contain points which not necessarily are desirable. As example, a point on the back of the bunny consider neighbors on its ear. However, those points have small influence on the tensor estimation, and some of them are disconsidered through the ϕ_{max} angle constraint.

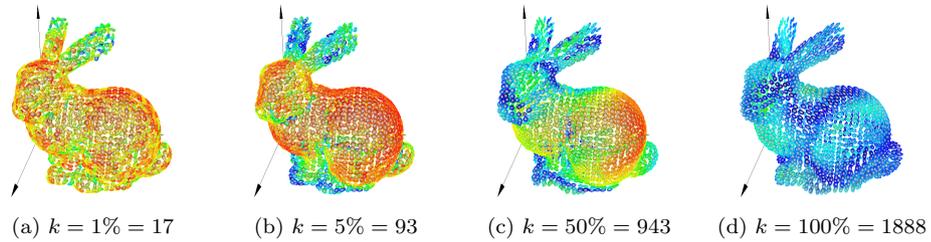


Fig. 3: (a)-(d): Orientation tensors for the *Bunny* point cloud, with different values of k , set as percentuals of the number of points. Tensors highlighted in hotter colors have higher planarity coefficients. Note that as k gets bigger, the number of red tensors (i.e. with higher c_p) diminish.

Our formulation of considering σ_p dependent of the k -th nearest neighbor makes the method sensitive to the density of the point cloud. If this density is not homogeneous, σ_p will have different values on different regions of the point cloud. A decay based on a distance threshold can produce more uniform results in such cases, but this would require a parameter calibration for each point cloud, which would be highly sensitive to its scale. The use of a percentual value of the number of points makes easier the parameter tuning.

2.2. Comparative Tensor Shape Factor

After the computation of the orientation tensors, we have a local estimative of the neighborhood distribution for each point of the mesh. Points that have a similar neighborhood disposition tend to have associated tensors with similar eigenvalues. Therefore, if we can measure dissimilarity between eigenvalues, we can express numerically the dissimilarity between the neighborhood distribution of two points, and this can be used to enhance the quality of correspondences between two meshes.

We define a factor to compare the normalized shapes called Comparative Tensor Shape Factor (CTSF). A second-order tensor can be geometrically represented as an ellipsoid, in which the magnitude of its axes is proportional to the eigenvalues of

the covariance matrix. Thus, two ellipsoids have the same shape if the proportion of their axes, i.e. matrix eigenvalues, is the same.

The tensors \mathbf{S}_i obtained in the curvature estimation step are normalized using the L_2 matrix norm, obtaining $\hat{\mathbf{S}}_i$. By normalizing the tensors of both Model and Data point sets, we can compare the local geometry despite the densities involved. As a consequence, the difference of magnitude of the original tensors is irrelevant in our method.

The CTSF between two tensors \mathbf{S}_1 and \mathbf{S}_2 is given by:

$$CTSF(\mathbf{S}_1, \mathbf{S}_2) = \sum_{k=1}^3 \left(\lambda_k^{\hat{\mathbf{S}}_1} - \lambda_k^{\hat{\mathbf{S}}_2} \right)^2$$

where $\lambda_k^{\hat{\mathbf{S}}_m}$ is the k -th greatest eigenvalue of the tensor $\hat{\mathbf{S}}_m$. Bigger values of CTSF indicate tensors with dissimilar shapes.

The eigenvalues of a tensor are invariant to rigid transformations, due to the isometric nature of these transformations. Therefore, this measure is suitable to the ICP matching step as a compatibility factor between two points. Table 1 shows some cases of high and low CTSF between two tensors.

Table 1: Examples of how the CTSF is affected by the geometry of planar tensors. Note that the CTSF is invariant to the orientation of the tensors and to their magnitude, due to the normalization.

Low CTSF			High CTSF		
					

2.3. Modified Iterative Closest Point

Assuming that both point sets of the ICP are different visions from the same object or surface and that the sampling rate of the sets is the same, the geometric neighborhood disposition of a point $m_i \in M$ is the same of its correspondent $d_j \in D$. Thus, the tensor \mathbf{S}_{m_i} has the same shape of \mathbf{S}_{d_j} , and the CTSF between them tends to zero.

The modification on the ICP proposed by this work aims to benefit from the invariance to rigid transformations of the CTSF in order to provide better correspondences on situations where the closest point provided by euclidean distance can be inaccurate. However, if only shape information is used in the matching step, the alignment obtained is coarse, and small details on the adjustment of the sets

are compromised. Thereby, we consider the distance in the matching step as a combination of the CTSF and the euclidean distance, using a weighting strategy to control their relative influence.

Our method uses a weighting factor w to combine CTSF and euclidean distance (ED), and controls the variation of the parameter when the algorithm reaches a local optimum of the error function. Let i be the number of local optima reached. We define the matching function as:

$$d(p, q) = ED(p, q) + w_i \cdot CTSF(\mathbf{S}_p, \mathbf{S}_q),$$

$$w_i = w_0 \cdot b^i, b < 1 \text{ and } 0 \leq w_i < w_0$$

where the parameter b impacts on the variation of the relative weight between CTSF and euclidean distance at each ICP loop, and w_0 is the initial influence of the euclidean distance, most likely a small value. In practice, the weight of the CTSF is divided by b every time a local optimum is reached. The algorithm stops when $w_i = \epsilon_2$, $\epsilon_2 \approx 0$, so that only euclidean distance is considered on its last loop. With this process we are able to guide the solution in order to give more importance to the tensor dissimilarity in the first iterations, coarsely recovering the transformation for higher angle displacements, and to reach fine alignment when the sets are close enough, condition in which the matching based on euclidean distance (original ICP formulation) tends to work better. We set $w_0 = 10000$, $\epsilon_2 = 10^{-6}$ and $b = 0.1$ as default values of the parameters, but show results with different values of b . Our method is summarized on Algorithm 1.

Algorithm 1 Modified Version of the ICP algorithm

Require: M, D, k, a, w_0

- 1: Tensor Estimation of M
 - 2: Tensor Estimation of D
 - 3: $i \leftarrow 0$
 - 4: $w_i \leftarrow w_0$
 - 5: **while** $w_i > \epsilon_2$ **do**
 - 6: $C \leftarrow$ Nearest Neighbors($M, D, w_i, 1 - w_i$)
 - 7: $T \leftarrow$ Transformation Estimation (D, C)
 - 8: **if** $\text{RMS}(T \cdot D, M) < \text{RMS}(D, M)$ **then**
 - 9: update T
 - 10: **else**
 - 11: $i \leftarrow i + 1$
 - 12: $w_i \leftarrow b^{(\log_a w_0) - i}$
 - 13: **end if**
 - 14: **end while**
 - 15: **return** T
-

The CTSF and the euclidean distance are measurements of distinct magnitudes,

since one of them is a distance factor and the other one is a shape dissimilarity factor.

In the optimization aspect, each step of the weight variation is an execution of the ICP algorithm with a different relative weight associated to the CTSF on the matching of points. The weight variation works as a coarse-to-fine successive approximation method. Each step of the process is a scale reduction of the CTSF in comparison to the euclidean distance.

The value of b controls the reduction on the weight factor of each step of the process. With smaller values, the method perform fewer steps, and in most cases fewer iterations. For values near 1, the method performs more iterations and possibly reaches better results, since the landscape of the function is changed more smoothly. Therefore, an increase on b implies on an increase of the precision of the transformation estimation and bigger chances of convergence. The downside is that a linear increase on the value of b implies on an exponential increase on the number of iterations in this formulation.

Since our algorithm alters the matching step of the Iterative Closest Point, it can be used alongside any minimization strategy. In our experimental setup, we modify both the original ICP and the Sparse ICP³ to use the CTSF in the matching step, combined with the weight variation strategy.

3. Experimental Setup

In rigid registration, it is not usual to make a quantitative evaluation of the methods, since too much time is spent on the execution on large point clouds. Instead, the most common is to validate them through qualitative comparison. This kind of analysis is generally inconclusive, because the behavior observed may not describe the general performance. To achieve a level of statistical relevance, we propose an experimental setup composed by a large base of events capable of quantitatively expressing the behavior of the methods in a variety of situations.

The dataset used in our work is composed by synthetically generated events that deal with usual issues on rigid registration. We consider an event as an alignment situation between two point clouds. Our dataset has two types of events: the first type contains point clouds with outliers and additive noise and the second deals with partially overlapped point clouds, also synthetically generated.

The base point clouds used in the generation of the events were sampled versions of the Bunny^a, the Happy Buddha^a, the Octopus^b and the Genus-2^c. For the Bunny point set we use the smallest zippered version available. The other sampled point sets were obtained through a Poisson-disk sampling algorithm³⁵ available

^aProvided by Stanford University Computer Graphics Laboratory on <http://graphics.stanford.edu/data/3Dscanrep/>.

^bProvided courtesy of INRIA by the AIM@SHAPE-VISIONAIR Shape Repository.

^cProvided by École Polytechnique Fédérale de Lausanne Computer Graphics and Geometry Laboratory on http://lgg.epfl.ch/statues_dataset.php

on MeshLab^d. This sampling process was necessary due to the large number of executions performed on the experimental setup.

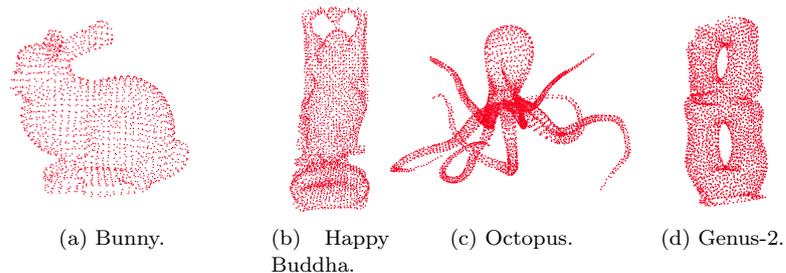


Fig. 4: The point sets used on the experiment: (a): Bunny (containing 1889 points). (b): Happy Buddha (3118 points). (c): Octopus (3822 points). (d): Genus-2 (2711 points).

Those point sets were chosen because they have different geometric features: the majority of points of the Bunny have a smooth neighborhood, except the points on the ears and on the feet, which are high curvature regions. The Happy Buddha also have many smooth patches, but presents symmetry on the vertical and horizontal axes. The Octopus, on the other side, has more high curvature patches in its tentacles. The Genus-2 is a point cloud composed mostly by planar regions, but can be challenging due to its symmetry axes, increasing the chance of misalignment. Figure 4 shows the point clouds that compose our dataset.

We scale the point sets to a bounding box whose biggest side is 1.0, keeping the original aspect ratio. This normalization is the first step of the event generation and is applied before the rigid transformation, the addition of outliers and additive noise and the generation of the subsets on the partial overlapping case. This way, the error measurements for all the point clouds are on a similar order of magnitude.

For wide angle result evaluation, we sample the rotation angle interval $[0^\circ - 180^\circ]$ at each 15° . The dataset has the same number of events for each angle and parameter configuration. In each event, an axis-angle rotation is applied in one of the point clouds, through the generation of a random unit vector \vec{v} following an isotropic distribution in an unit sphere. One of the meshes is rotated around \vec{v} . The transformation does not include a translation part. We also consider this a more challenging scenario for the ICP, since there is a bigger chance of reaching a wrong local minima, specially on wide angles where the sets are incorrectly prealigned.

Additive noise, outliers and partial overlapping are generated randomly by synthetic processes, with a different seed for each process. All the random variables

^d<http://meshlab.sourceforge.net>

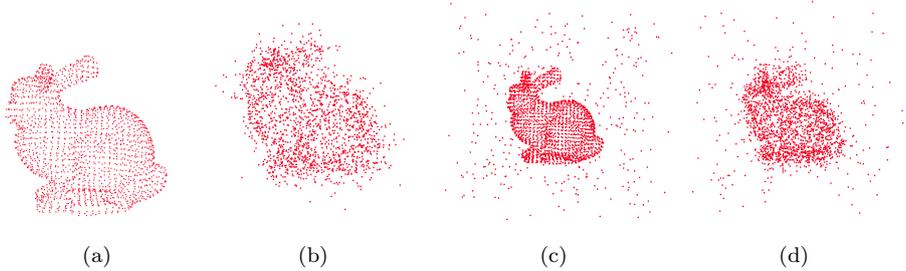


Fig. 5: Examples of application of additive noise and outliers to the Bunny point set. (a): original Bunny. (b) Bunny with additive noise ($\delta = 0.10$). (c) Bunny with 20% of Outliers. (d) Bunny with additive noise and outliers combined ($\delta = 0.10$ and 20% of outliers).

are generated by a Mersenne Twister pseudo-random number generator. The next Subsections detail the synthetic algorithms for generation of events with additive noise, outliers and partial overlapped point clouds. The dataset guarantees that the results generated are paired, i.e. the same events are executed for all methods.

3.1. Additive Noise and Outliers

The first type of event of the dataset addresses realistic non-optimal scenarios with additive noise and outliers. The point sets are fully overlapped, that is, all the points on the base mesh are used to generate both Model and Data meshes, which are distinguished by the application of a rigid transformation and the addition of outliers and additive noise on both.

Outliers are randomly generated by a uniform distribution in a sphere with radius 2.0, with the biggest side of the normalized bounding box having size 1.0. For additive noise, we perturb each point \vec{p}_i adding a vector \vec{r}_i , obtained by an isotropic distribution over an unit sphere, scaled by a factor δ sampled using a Gaussian distribution: $\vec{p}_i = \vec{p}_i + \delta \cdot \mathcal{N}(0, 1) \cdot \vec{r}_i$.

This scheme of synthetic additive noise generation is used since it is a harder scenario compared to the addition of a perturbation on the normal direction of each point. We generate events with a variable amount of outliers, 0%, 5% and 20% of the number of points on the mesh, and a variable perturbation factor δ for the additive noise: 0, 0.01 and 0.05. Higher amounts of outliers and additive noise does not represent realistic situations. For δ values bigger than 0.05, the additive noise degenerates the mesh such that most of its geometrical information is lost. Figure 5 examples the effect of additive noise and outliers.

3.2. Partial Overlapping

The second type of event addresses partial overlapping situations, simulating the process of object reconstruction. For that, we generate subsets of the base meshes that are continuous in relation to the nearest neighbors list, controlling the percentage amount of non-overlapping (α) and overlapping points (β) between those subsets, relative to the number of points on the base mesh. Table 2 shows the parameter combinations for α and β .

Table 2: Variation of the parameters for the second type of event: percentage of points on the overlapping region (α) and on the non-overlapping region (β).

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

The algorithm for synthetic subset generation is a breadth search on the k-nearest-neighbors list, initiated on a random point to generate the overlapping region, then generating the non-overlapping regions through a breadth search on points on the boundary of the overlapping region until the number of points desired is reached.

As the rate between the overlapping points and the non-overlapping points of the meshes diminish, it is expected to be harder for the methods to align the point sets. Like the first type of event, we also sample the rotation angle interval at each 15° .

Figure 6 shows examples of partial overlapping events.

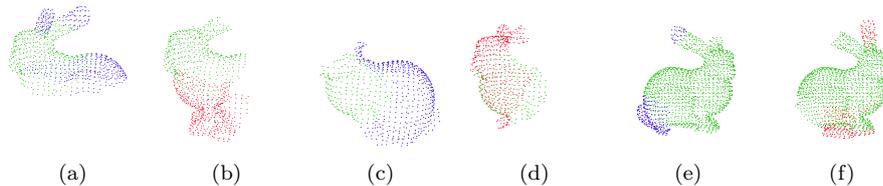


Fig. 6: Partial overlapping examples: green points indicate the overlapping region, and blue and red points indicate the non-overlapping region. (a) and (b): $\alpha = 25\%$, $\beta = 25\%$, average difficulty. (c) and (d): $\alpha = 12.5\%$, $\beta = 25\%$, the hardest parameter setup, since the overlapping region is smaller than the non-overlapping regions. (e) and (f): $\alpha = 12.5\%$, $\beta = 75\%$, the easiest parameter setup, since the overlapping region is big.

4. Experimental Results

4.1. *Error Measurement Protocol*

In the fine registration literature, many error measurements are presented, such as residual errors, convergence regions and rotation and translation error. Since most works present qualitative results, visual results are shown in order to link these error measurements with the convergence of the methods on the chosen examples. Many works also present data on the number of iterations and the time spent on the execution, in order to highlight their computational efficiency.

In this work, we use two types of error measurements: the ground-truth root mean squared error (GT-RMS), which is the average euclidean distance between the points on the first point cloud and their correct correspondences on the second, and the labeled error, which is the number of correct correspondences between both clouds.

The GT-RMS error is only measured between inliers, since outliers do not have correct correspondences. For the same reason, in partial overlapping events, only points on the overlapping region are considered. As a consequence, this error measurement can only be used with synthetically generated point clouds. It is important to emphasize the difference between the RMS used on the ICP, measured between the point and the correspondence estimated by the method, and the GT-RMS, between the point and its known correct correspondence. The GT-RMS is used only for error evaluation and cannot be used in the ICP, since it requires a ground-truth of previously known correspondences.

The labeled error can be used to evaluate cases of exact correspondences, since there is a guarantee that each point has a correspondent on the other mesh. In additive noise situations, the exactness of this measurement is lower, because the original position of the points is modified.

For the quantitative analysis, we have two clear result patterns - success and failure. If an algorithm fails to reach alignment on an event, it does not matter whether the final error obtained is big or not. Mean and standard deviation of the GT-RMS can be affected by the magnitude of the error in failure cases, and this fact can lead to wrong conclusions about the method performance. Therefore, the convergence rate can be better analyzed through a success/failure histogram. As a global success rate, we count the number of successful experiments through a threshold segmentation on the GT-RMS error and the labeled error, whose limit values are set based on previous qualitative observations and depend of the amount of additive noise and outliers.

For events with fully overlapped point clouds, the threshold criterion considered is simple on situations without additive noise: 95% of the correspondences must be correct, and the GT-RMS error is at maximum 10^{-2} . Outliers would not alter the GT-RMS criterion neither the labeled error, so the threshold criterion stands the same.

As stated before, additive noise can affect the exactness of the error measure-

ments. The number of correct labeled correspondences tends to decay even on successful cases, but it is still possible to distinguish success from failure by those thresholds. In these cases, we define a minimal labeled error of 100 correct correspondences combined with a maximal RMS error of 0.1, so that it is highly unlikely that a failure event is not identified by those threshold values. However, with high levels of additive noise, this threshold segmentation becomes harder, and some successful events might be classified as a failure. Table 3 shows the threshold values considered on our evaluation.

Table 3: Threshold values used for events with additive noise and outliers

Situation	Max GT-RMS	Min labeled
Without additive noise	10^{-2}	$N_p \cdot 95\%$
With additive noise	10^{-1}	100 points

On the partial overlapping events, both labeled error and RMS error were calculated considering only the ground-truth correspondences of the overlapping region. We consider an event successful when its GT-RMS error is lower than a threshold of 0.05 and it has more than 90% of the points on the overlapping region with correct labeled correspondences. The list of ground-truth correspondences is computed on the subset generation algorithm.

4.2. Results with additive noise and outliers

To characterize the behavior of our method with different parameter values, we separate 15 events of the first type for each value of angle, additive noise and outliers, performing a preliminary step with 36 combinations of method parameters, i.e. the size of the nearest neighbor list (k), the ellipsoid angle α_{ellip} and the cut angle ϕ_{max} . The parameter b is set as 0.1 in this step, since this parameter is independent from the others, and its behavior is known: bigger values reaches better results performing more iterations. We use a value that is a midterm between performance and time spent.

In this preliminary step, the correspondences are always established using the CTSF-based matching and the transformation is estimated using Horn quaternion method ³⁶. The difference between the methods is in the information encoded on tensors that are used as input for the ICP.

Table 4 shows the parameter configurations evaluated on the preliminary step.

Thereupon, we use the parameter values that presented better results and execute more 30 events to compare the behavior of our method with other approaches. We raise the value of the parameter b to 0.75 in order to achieve better results, since the amount of methods to be executed is lower. In this step, we execute the original ICP and the Sparse ICP ³ (here shortened S ICP), with both matching functions based on CTSF and euclidean distance.

Table 4: Combination of parameters α_{ellip} and ϕ_{max} . For each pair, we execute 15 events with different values of k : 1%, 5%, 25%, 50%, 75%, 100%

α_{ellip}	30°	30°	45°	30°	45°	60°
ϕ_{max}	36°	45°	45°	60°	60°	60°

The first step of the experiment is not executed with the Sparse ICP because it would yield a high computational cost. Yet, since all the parameters evaluated affect only the tensor estimation, the correspondences provided by the CTSF are the same. Thus, parameters that yield better results on the Original ICP with the CTSF should also reach better results on other methods using the CTSF.

For the Sparse ICP, we use the norm parameter $p = 0.4$, which offers a good trade-off between performance and robustness, according to its authors. The error sequence of the Sparse ICP, unlike the original ICP, is not monotonically descent, so the standard method stops when a fixed number of ICP iterations is made, whose default value is 100, or when the error reaches below a stopping threshold. In order to adapt the CTSF weighting scheme to the Sparse ICP method, for each weight value we execute 100 iterations of the method or until the error is smaller than the threshold parameter. The decay of the weight parameter is the same as described in Algorithm 1. The stopping threshold error parameter is set to 10^{-5} . The other parameters are the default parameters of the source code made available by the authors^e.

4.2.1. *First Step - Best parameter search*

In the first step of the experimental setup for outliers and additive noise we highlight the parameter combination with the best overall performance for each point cloud, and some of the observed tendencies. We perform 15 experiments for each configuration, although ideally more results should be generated in order to obtain better statistical backup.

The results are grouped by the presence/absence of outliers and additive noise. The last row is the overall success rate of the method. Since the experiments are paired, the same conditions were tested for all the methods. Table 5 shows for each point cloud and value of k the best average success rate of the combinations in Table 4.

In general, for all point clouds, higher values of k lead to better results with outliers. It is intuitive that the influence of an outlier will be attenuated as the neighborhood gets larger. Although larger neighborhoods include either outliers and inliers, the coplanarity induced by the second step of the tensor estimation process makes the tensors encode effectively the surface geometry and attenuates the influence yielded by neighbors misaligned to the tangent plane. This way, tensors

^eAvailable at <https://code.google.com/p/sparseicp/>

Table 5: Results with additive noise and outliers.

Bunny							
k	α_{ellip}	ϕ_{max}	Clean	Noise	Outliers	Noise + outliers	Overall
100.00%	60°	45°	100.00%	100.00%	100.00%	97.78%	99.01%
75.00%	45°	45°	100.00%	100.00%	100.00%	100.00%	100.00%
50.00%	45°	30°	100.00%	100.00%	100.00%	99.86%	99.94%
25.00%	60°	30°	100.00%	100.00%	99.17%	96.94%	98.46%
5.00%	60°	45°	100.00%	88.89%	69.44%	55.97%	71.17%
1.00%	60°	60°	100.00%	79.44%	48.61%	35.42%	55.31%
Happy Buddha							
k	α_{ellip}	ϕ_{max}	Clean	Noise	Outliers	Noise + outliers	Overall
100.00%	60°	60°	100.00%	99.17%	99.44%	93.61%	96.85%
75.00%	45°	30°	100.00%	90.83%	98.89%	86.53%	91.73%
50.00%	60°	60°	100.00%	100.00%	98.61%	87.92%	94.32%
25.00%	60°	30°	100.00%	87.78%	77.50%	57.08%	73.21%
5.00%	60°	30°	100.00%	85.28%	48.89%	33.61%	55.86%
1.00%	60°	60°	100.00%	95.28%	38.89%	29.03%	53.83%
Octopus							
k	α_{ellip}	ϕ_{max}	Clean	Noise	Outliers	Noise + outliers	Overall
100.00%	60°	45°	100.00%	90.56%	100.00%	93.33%	94.94%
75.00%	60°	45°	100.00%	94.17%	99.44%	90.00%	94.14%
50.00%	60°	30°	100.00%	90.83%	88.89%	76.39%	85.00%
25.00%	60°	30°	100.00%	90.83%	83.06%	68.89%	80.37%
5.00%	60°	60°	100.00%	93.61%	58.89%	52.36%	68.27%
1.00%	60°	60°	100.00%	95.28%	39.72%	32.27%	55.44%
Genus-2							
k	α_{ellip}	ϕ_{max}	Clean	Noise	Outliers	Noise + outliers	Overall
100.00%	60°	60°	100.00%	91.67%	86.11%	73.06%	83.09%
75.00%	60°	30°	100.00%	93.06%	83.89%	69.58%	81.36%
50.00%	60°	60°	100.00%	91.11%	82.50%	65.14%	78.64%
25.00%	60°	60°	100.00%	95.56%	72.22%	59.72%	74.94%
5.00%	60°	45°	100.00%	81.39%	49.72%	45.00%	60.25%
1.00%	60°	60°	100.00%	87.22%	47.50%	40.42%	59.01%

with higher values of k are more robust to the presence of outliers, and conversely achieve better results on the experiment.

Additive noise affects directly the structure of the surface. Therefore, tensors are likely to lose their planarity and the precision of their estimation becomes lower, regardless of the size of the neighborhood. In most cases, the performance was similar for all values of k , and even smaller values like $k = 1\%$ reached consistent results in cases with only additive noise. However, since those values are sensitive to outliers, the overall performance is worse, and bigger values of k are recommended for more robustness.

The angle parameters α_{ellip} and ϕ_{max} had only a slight influence on the results, such that there is no dominant parameter combination. The method is more sensitive to variations on the parameter k , especially for higher angle displacements.

The predominance of low-curvature regions and the absence of symmetry of the Bunny improve the accuracy of the correspondences generated by the CTSE, yielding better results than the other point clouds tested, with an overall performance

above 90% with $k \geq 25\%$. More specifically, $k = 75\%$ obtained better results and the overall performance of small values of k was negatively impacted by outliers. The chosen parameter combination for the second step was $k = 75\%$, $\alpha_{ellip} = 60^\circ$, $\phi_{max} = 60^\circ$, that yielded success in 100% of the events.

On the Happy Budha point cloud, high convergence rates were reached for $k \geq 50\%$. The performance in cases with only additive noise was not impacted, but lower values of k had problems dealing with outliers. The parameter combination chosen was $k = 100\%$, $\alpha_{ellip} = 60^\circ$, $\phi_{max} = 60^\circ$, with an overall convergence rate of 96.85%.

On the Octopus, the differences of performance between the values of k were more distinguishable in cases with outliers, in which bigger values of k reached better results. For cases with only additive noise, the performance was similar between the different parameters. The combination chosen was $k = 100\%$, $\alpha_{ellip} = 60^\circ$, $\phi_{max} = 45^\circ$, with overall performance of 94.94%.

The Genus-2 was the hardest point cloud of the dataset. The performance was strongly affected by outliers, impacting negatively the overall success rates. The best overall success rate was 83.09%, with $k = 100\%$, $\alpha_{ellip} = 60^\circ$, $\phi_{max} = 60^\circ$. This combination of parameters performed particularly well with outliers, reaching 86.11% with only outliers and 73.09% with both outliers and noise, smaller rates if compared to the results on the other point clouds.

4.2.2. *Second Step - Best result search*

In this step, we used the best parameters obtained on the first step for more 30 executions on each point cloud. The methods evaluated were the Original ICP and the Sparse ICP, with matching functions based on the CTSF and euclidean distance. The value of b was raised to 0.75, in order to enhance the results of the method. This also increases the execution time of the method, and the time spent on most executions of the Sparse ICP combined with the CTSF was very high, if compared to other methods.

For this step, we detail the success rate for each level of outlier and additive noise. The Tables 6, 7, 8 and 9 show the better performance of the CTSF methods, and Figures 7, 8, 9 and 10 show that the performance difference is bigger on high angles. This was expected, since such results are similar to the ones obtained on the first step for this parameter combination, and fine methods were not designed to cope with bad initialization situations. However, with small angles (up to 30°) the convergence rates are reasonable for the original methods with euclidean distance-based matching. This way, the results obtained are coherent, and with the proper coarse initialization the methods should be able to reach better convergence rates.

Table 6: Success per noise and outlier level - Bunny

ICP	0.00%	5.00%	20.00%	Overall
$\sigma = 0.00$	29.44%	29.44%	21.11%	26.67%
$\sigma = 0.01$	16.94%	21.94%	14.17%	17.69%
$\sigma = 0.05$	16.67%	18.61%	13.06%	16.11%
Overall	21.02%	23.33%	16.11%	20.15%
S ICP	0.00%	5.00%	20.00%	Overall
$\sigma = 0.00$	44.44%	45.56%	40.56%	43.52%
$\sigma = 0.01$	41.11%	40.28%	35.28%	38.89%
$\sigma = 0.05$	32.78%	33.61%	28.89%	31.76%
Overall	39.44%	39.81%	34.91%	38.06%
ICP-CTSF	0.00%	5.00%	20.00%	Overall
$\sigma = 0.00$	100.00%	99.72%	100.00%	99.91%
$\sigma = 0.01$	100.00%	100.00%	100.00%	100.00%
$\sigma = 0.05$	100.00%	100.00%	100.00%	100.00%
Overall	100.00%	99.91%	100.00%	99.97%
S ICP-CTSF	0.00%	5.00%	20.00%	Overall
$\sigma = 0.00$	100.00%	95.00%	100.00%	98.33%
$\sigma = 0.01$	100.00%	100.00%	100.00%	100.00%
$\sigma = 0.05$	100.00%	100.00%	98.33%	99.44%
Overall	100.00%	98.33%	99.44%	99.26%

Table 7: Success per noise and outlier level - Happy

ICP	0.00%	5.00%	20.00%	Overall
$\sigma = 0.00$	40.00%	40.56%	33.06%	37.87%
$\sigma = 0.01$	22.50%	30.00%	21.39%	24.63%
$\sigma = 0.05$	33.61%	36.39%	31.39%	33.80%
Overall	32.04%	35.65%	28.61%	32.10%
S ICP	0.00%	5.00%	20.00%	Overall
$\sigma = 0.00$	48.06%	49.17%	33.33%	43.52%
$\sigma = 0.01$	47.22%	47.22%	33.61%	42.69%
$\sigma = 0.05$	41.39%	43.06%	36.11%	40.19%
Overall	45.56%	46.48%	34.35%	42.13%
ICP-CTSF	0.00%	5.00%	20.00%	Overall
$\sigma = 0.00$	100.00%	100.00%	100.00%	100.00%
$\sigma = 0.01$	100.00%	100.00%	95.28%	98.43%
$\sigma = 0.05$	99.44%	98.61%	92.50%	96.85%
Overall	99.81%	99.54%	95.93%	98.43%
S ICP-CTSF	0.00%	5.00%	20.00%	Overall
$\sigma = 0.00$	100.00%	86.94%	75.00%	87.31%
$\sigma = 0.01$	100.00%	98.89%	76.67%	91.85%
$\sigma = 0.05$	91.11%	87.22%	73.33%	83.89%
Overall	97.04%	91.02%	75.00%	87.69%

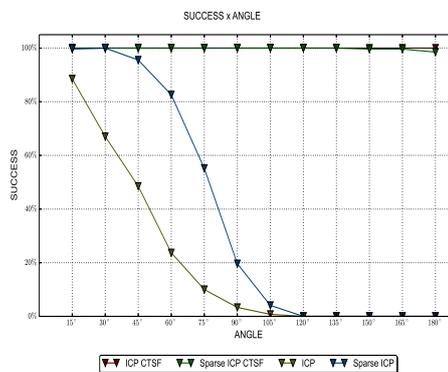


Fig. 7: Success per angle - Bunny

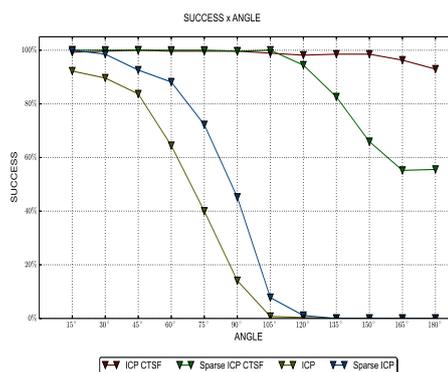


Fig. 8: Success per angle - Happy

Table 8: Success per noise and outlier level - Octopus

ICP	0.00%	5.00%	20.00%	Overall
$\sigma = 0.00$	14.44%	16.94%	8.89%	13.43%
$\sigma = 0.01$	11.39%	14.72%	6.11%	10.74%
$\sigma = 0.05$	9.72%	14.17%	6.11%	10.00%
Overall	11.85%	15.28%	7.04%	11.39%
S ICP	0.00%	5.00%	20.00%	Overall
$\sigma = 0.00$	26.39%	28.33%	29.44%	28.06%
$\sigma = 0.01$	26.11%	23.06%	26.39%	25.19%
$\sigma = 0.05$	20.00%	24.44%	25.28%	23.24%
Overall	24.17%	25.28%	27.04%	25.49%
ICP CTSF	0.00%	5.00%	20.00%	Overall
$\sigma = 0.00$	100.00%	100.00%	100.00%	100.00%
$\sigma = 0.01$	96.94%	100.00%	100.00%	98.98%
$\sigma = 0.05$	90.56%	98.32%	94.66%	94.50%
Overall	95.83%	99.44%	98.23%	97.83%
S ICP CTSF	0.00%	5.00%	20.00%	Overall
$\sigma = 0.00$	100.00%	99.17%	95.28%	98.15%
$\sigma = 0.01$	100.00%	98.61%	92.44%	97.03%
$\sigma = 0.05$	94.17%	91.94%	89.66%	91.93%
Overall	98.06%	96.57%	92.47%	95.70%

Table 9: Success per noise and outlier level - Genus-2

ICP	0.00%	5.00%	20.00%	Overall
$\sigma = 0.00$	36.94%	36.11%	34.72%	35.93%
$\sigma = 0.01$	30.56%	32.50%	23.89%	28.98%
$\sigma = 0.05$	31.01%	29.72%	31.94%	30.89%
Overall	32.84%	32.78%	30.19%	31.93%
S ICP	0.00%	5.00%	20.00%	Overall
$\sigma = 0.00$	50.83%	51.94%	50.28%	51.02%
$\sigma = 0.01$	45.83%	48.06%	49.17%	47.69%
$\sigma = 0.05$	39.44%	39.44%	43.06%	40.65%
Overall	45.37%	46.48%	47.50%	46.45%
ICP CTSF	0.00%	5.00%	20.00%	Overall
$\sigma = 0.00$	100.00%	99.17%	82.78%	93.98%
$\sigma = 0.01$	99.44%	95.83%	76.67%	90.65%
$\sigma = 0.05$	90.25%	81.11%	65.28%	78.87%
Overall	96.57%	92.04%	74.91%	87.84%
S ICP CTSF	0.00%	5.00%	20.00%	Overall
$\sigma = 0.00$	99.17%	63.31%	56.39%	72.98%
$\sigma = 0.01$	71.39%	63.33%	58.31%	64.37%
$\sigma = 0.05$	65.27%	61.11%	55.83%	60.72%
Overall	78.64%	62.58%	56.84%	66.03%

The Sparse ICP with the CTSF obtained poor convergence rates on the Genus-2 point cloud, which was pointed out in the first step as the hardest point cloud of the dataset. The graphic shows that the Sparse ICP was affected by high angles, probably because of the number of a local optima easily reachable from angles bigger than 90° .

4.3. Results with partial overlapping

In this type of event, we evaluate 30 events for each parameter configuration of angle and overlapping/non-overlapping amount. The methods chosen were the Sparse ICP method, the original ICP, and its trimmed version, using match functions based on the euclidean distance and on the CTSF. As stated before, it is known that the original ICP formulation is not proper for partially overlapped point clouds, therefore its performance tends to be poor.

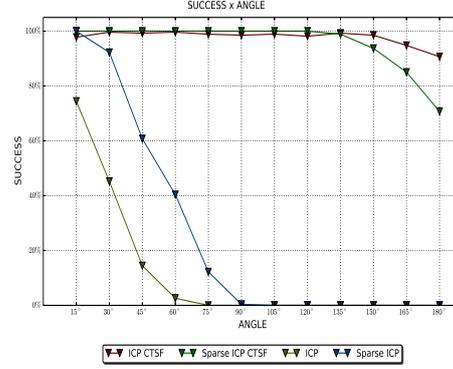


Fig. 9: Success per angle - Octopus

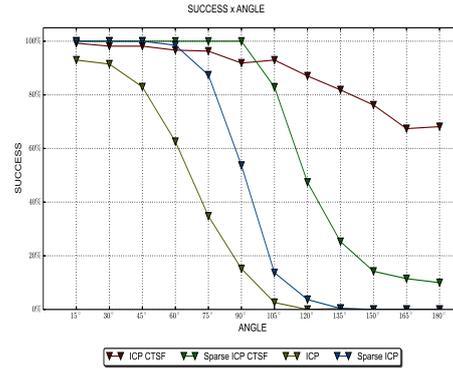


Fig. 10: Success per angle - Genus-2

In this case, we chose not to do the preliminary step and use the values of $\alpha_{ellip} = 45^\circ$ and $\phi_{max} = 45^\circ$, due to the high computational time that would be spent.

Specifically on partially overlapped point clouds, only correspondences between points over the common region are considered correct. The effectiveness of the CTSF is possibly lower, because the neighborhood of the points on the overlapping region can include points outside of this region, generating different tensors for each point cloud. As the value of k increases, the difference between the regions represented by the tensors would also increase, since points outside of the overlapping region have a bigger chance of being used in the computation of the tensor. Conversely, smaller values of k have a tendency to provide similar tensors between correspondences on the overlapping region, since the number of neighbors outside this region tends to be smaller and limited to a boundary of the region. If the size of the neighborhood is too small, however, the information cast on the tensors can be insufficient to represent the local geometry. We test four values of k : 15%, 10%, 5% and 1% of the points.

The Trimmed ICP was included on the evaluation of this type of event since it is a classical strategy to deal with partial overlapping point sets. For this method, we fixed the amount of discarded points in 10% of the number of points of the mesh, although ideally the amount should be proportional to the overlapping percentage. In practical situations, the amount of overlapping between the point sets usually is unknown. Therefore, this value is set as an intermediate value.

Table 10 shows the percentual convergence rate per overlapping and non-overlapping amount.

In this experiment, it is clear that the CTSF enhances the convergence of the ICP also in partial overlapping situations. In the large majority of situations, the CTSF-based methods obtained better results than the original methods. However, the success rates were nowhere near the ones obtained on full overlap events, and there is a substantial decay as the proportion between the number points over the overlapping region and over the non-overlapping region gets smaller.

The Sparse ICP with the CTSF presented the best overall performance with partial overlapping. All the CTSF-based methods presented a slight variation with relation to the rotation angle, which indicates that the CTSF is effective for high angle displacements even in partial overlapping situations, yielding approximately the same convergence probability for all angle values. However, most of the parameter combinations for partial overlapping had proven themselves as hard scenarios, with very low convergence rates. The Sparse ICP had an advantage in comparison to the Trimmed ICP in this aspect, reaching much better rates particularly in cases where $\alpha = 25\%$, for which the Trimmed ICP was hardly successful. The original ICP method combined with the CTSF achieved good results for $\alpha = 12.5\%$ and $\beta = 75\%$ on the Bunny and Happy Budha, but in general performed poorly.

As expected, smaller values of k presented slightly better results, with the exception of the Bunny. This tendency is more easily observed on cases with a low over-

Table 10: Partial Overlapping results

Bunny									
α	12.5%				25.0%				Overall
β	75.0%	50.0%	25.0%	12.5%	50.0%	37.5%	25.0%	12.5%	
ICP-CTSF $k = 15\%$	91.39%	95.00%	16.39%	00.00%	45.83%	04.72%	00.28%	00.00%	31.70%
ICP-CTSF $k = 10\%$	95.00%	95.83%	16.39%	00.00%	50.28%	05.28%	00.28%	00.00%	32.88%
ICP-CTSF $k = 5\%$	93.89%	89.44%	19.72%	00.28%	50.00%	05.56%	00.00%	00.00%	32.36%
ICP-CTSF $k = 1\%$	75.83%	78.06%	21.94%	00.28%	44.44%	08.06%	00.00%	00.00%	28.58%
ICP	18.61%	16.67%	01.67%	00.00%	06.39%	01.39%	00.00%	00.00%	05.59%
S ICP-CTSF $k = 15\%$	93.89%	94.72%	72.22%	26.11%	84.12%	73.06%	50.70%	00.00%	61.85%
S ICP-CTSF $k = 10\%$	95.56%	96.94%	73.33%	27.50%	87.19%	80.78%	56.55%	00.00%	64.72%
S ICP-CTSF $k = 5\%$	95.56%	93.33%	75.56%	33.89%	89.69%	78.33%	55.00%	00.00%	65.16%
S ICP-CTSF $k = 1\%$	93.33%	93.89%	71.31%	32.22%	89.94%	75.21%	58.89%	00.00%	64.33%
S ICP	41.67%	29.44%	13.65%	06.11%	29.05%	17.83%	09.44%	00.00%	18.39%
Happy									
α	12.5%				25.0%				Overall
β	75.0%	50.0%	25.0%	12.5%	50.0%	37.5%	25.0%	12.5%	
ICP-CTSF $k = 15\%$	76.39%	80.00%	07.50%	00.00%	06.67%	00.28%	00.00%	00.00%	21.35%
ICP-CTSF $k = 10\%$	93.06%	85.00%	07.78%	00.00%	06.39%	00.00%	00.00%	00.00%	24.03%
ICP-CTSF $k = 5\%$	95.28%	88.33%	11.39%	00.00%	08.33%	00.28%	00.00%	00.00%	25.45%
ICP-CTSF $k = 1\%$	99.17%	93.33%	15.83%	00.00%	09.72%	00.28%	00.00%	00.00%	27.29%
ICP	31.11%	20.28%	01.39%	00.00%	01.67%	00.00%	00.00%	00.00%	06.81%
S ICP-CTSF $k = 15\%$	78.21%	82.45%	48.75%	01.39%	73.45%	53.61%	21.39%	00.00%	44.81%
S ICP-CTSF $k = 10\%$	89.89%	88.02%	53.20%	02.50%	77.12%	56.67%	27.78%	00.00%	49.27%
S ICP-CTSF $k = 5\%$	92.66%	91.34%	57.38%	06.94%	89.17%	64.72%	32.22%	00.00%	54.09%
S ICP-CTSF $k = 1\%$	94.25%	94.40%	55.99%	09.44%	91.27%	71.94%	39.17%	00.00%	56.80%
S ICP	46.57%	34.73%	10.86%	01.67%	33.61%	20.28%	07.22%	00.00%	19.26%
Octopus									
α	12.5%				25.0%				Overall
β	75.0%	50.0%	25.0%	12.5%	50.0%	37.5%	25.0%	12.5%	
ICP-CTSF $k = 15\%$	81.11%	58.06%	02.22%	00.00%	04.44%	00.00%	00.00%	00.00%	18.23%
ICP-CTSF $k = 10\%$	87.78%	57.78%	02.22%	00.00%	04.44%	00.00%	00.00%	00.00%	19.03%
ICP-CTSF $k = 5\%$	90.28%	63.61%	02.50%	00.00%	03.89%	00.00%	00.00%	00.00%	20.03%
ICP-CTSF $k = 1\%$	92.78%	75.28%	03.89%	00.00%	03.61%	00.00%	00.00%	00.00%	21.94%
ICP	14.17%	8.33%	00.56%	00.00%	00.28%	00.00%	00.00%	00.00%	02.92%
S ICP-CTSF $k = 15\%$	98.33%	74.72%	25.83%	02.50%	62.78%	42.50%	19.72%	00.00%	40.80%
S ICP-CTSF $k = 10\%$	99.17%	72.50%	23.89%	03.06%	73.61%	43.61%	21.94%	00.00%	42.22%
S ICP-CTSF $k = 5\%$	99.44%	75.56%	31.67%	02.22%	72.50%	49.44%	24.15%	00.00%	44.43%
S ICP-CTSF $k = 1\%$	99.72%	79.17%	28.61%	03.06%	83.33%	76.39%	36.54%	00.00%	50.89%
S ICP	26.67%	23.33%	10.28%	00.83%	20.56%	20.00%	10.83%	00.00%	14.06%
Genus-2									
α	12.5%				25.0%				Overall
β	75.0%	50.0%	25.0%	12.5%	50.0%	37.5%	25.0%	12.5%	
ICP-CTSF $k = 15\%$	77.22%	77.78%	15.56%	00.28%	12.50%	00.28%	00.00%	00.00%	22.95%
ICP-CTSF $k = 10\%$	82.50%	81.67%	18.33%	00.28%	15.00%	00.00%	00.00%	00.00%	24.72%
ICP-CTSF $k = 5\%$	83.06%	84.17%	22.50%	00.28%	15.28%	00.00%	00.00%	00.00%	25.66%
ICP-CTSF $k = 1\%$	83.89%	83.61%	25.56%	00.28%	17.22%	00.00%	00.00%	00.00%	26.32%
ICP	27.22%	19.17%	03.06%	00.00%	05.28%	00.00%	00.00%	00.00%	06.84%
S ICP-CTSF $k = 15\%$	75.83%	76.94%	52.50%	02.50%	71.11%	66.94%	26.11%	00.00%	46.49%
S ICP-CTSF $k = 10\%$	85.56%	83.61%	56.11%	03.06%	79.72%	75.28%	34.72%	00.00%	52.26%
S ICP-CTSF $k = 5\%$	90.28%	88.06%	61.11%	07.78%	86.11%	83.89%	43.06%	00.00%	57.53%
S ICP-CTSF $k = 1\%$	91.67%	83.89%	61.67%	13.33%	88.89%	82.78%	53.33%	00.00%	59.44%
S ICP	47.50%	38.06%	19.17%	03.89%	38.06%	28.89%	11.67%	00.00%	23.40%

lapping amount. The reduction of the number of points on the overlapping region significantly impacted both Sparse ICP-CTSF and Trimmed ICP-CTSF methods. On Table 10, we observe that the convergence rates were particularly lower when $\beta = 12.5\%$, and null in cases where $\beta = 12.5\%$ and $\alpha = 25\%$. Besides the natural difficulty of events with small overlapping regions, the CTSF also loses part of its accuracy in these cases, because the neighborhood used for the tensor estimation is hardly similar on points on both meshes.

Figure 11 shows a successful case of partial overlapping alignment for the Genus-2.

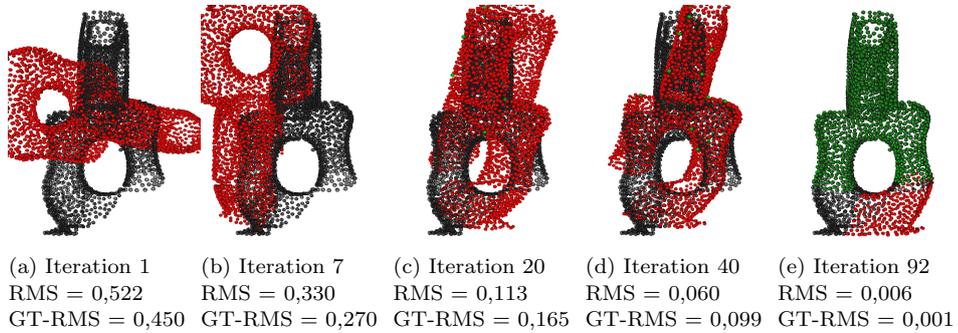


Fig. 11: Convergence for the Genus-2, partial overlapping with $\alpha = 75\%$, $\beta = 12.5\%$, rotation of 105° . Method: Trimmed ICP+CTSF, with $k = 1\%$, $\alpha_{ellip} = 60^\circ$, $\phi_{max} = 45^\circ$. Green points indicate correct correspondences.

5. Conclusion

This work presents a new method for rigid registration, based on tensor eigenvalues. This is the first rigid registration approach that treats covariance matrices as second-order tensors. The tensors are estimated by a two-step voting process using 3D tensor structuring elements, aiming to infer how likely their neighborhood form a surface. In order to use of the tensor information on the ICP, we define the CTSF, a similarity factor based on tensor invariant features. This factor is used on the ICP matching step in order to enhance the quality of correspondences. An heuristic weighting strategy between the euclidean Distance and the CTSF is proposed to guide the solution from a coarse alignment, based on tensor dissimilarity, to fine, based on euclidean distance. This enhances the convergence probability, specially on wider initial angle situations. Since only the matching step is modified, our approach can be used alongside many minimization methods. The major drawback is the preprocessing time needed for the tensor estimation, specially on large point sets.

The analysis made on our dataset is quantitative and considers only the success or failure of each event. On that sense, our experiments reveal that the convergence rate of the CTSF methods is enhanced in situations of wider angles, additive noise and outliers when compared to their respective non-CTSF versions. On partial overlapping situations, the CTSF was capable of enhancing the convergence of the Trimmed ICP and the Bouaziz's method in general. The wide angle registration of partial overlapping surfaces makes easier the task of 3D object reconstruction, thus the importance of such results.

The main parameter added to the ICP is the size k of the neighborhood of each point, which defines the extent of the structuring elements in the tensor estimation. It is important to note that no input point is discarded at any time during the process. Our method takes into account the information of all input points regardless they are inliers or outliers.

As future works, the CTSF can be used as a dissimilarity factor between any second-order tensors and applied in tasks other than rigid registration. Since our heuristic weights euclidean distance and dissimilarity between tensors, any covariance matrix that represents locally the surface can be used instead of ours. Other variants of the ICP can also be adapted to use the CTSF to enhance the alignment. Our tensor estimation step can be optimized in terms of speed, by computing the nearest neighbors list faster. It is also possible to determine a value of k that adapts better for each region of the mesh.

For more detailed results and discussions about the parameters of the method, we recommend refer to ³⁷. The dataset used on the experiment can be found at www.gcg.ufjf.br/datasets/icpctsf.zip.

Acknowledgment

We would like to thank CAPES and FAPEMIG for financial support.

References

1. P. J. Besl, N.D. McKay. "A method for registration of 3-D shapes". Pattern analysis and machine intelligence, IEEE Transactions on 14.2 (1992), p. 239-256.
2. M. B. Vieira, P. Martins, A. Araujo, M. Cord, S. Philipp-Foliguet, "Smooth surface reconstruction using tensor fields as structuring elements". Computer Graphics Forum, Vol. 23, Wiley Online Library, 2004, p. 813-823.
3. S. Bouaziz, A. Tagliasacchi, M. Pauly, "Sparse iterative closest point", Computer Graphics Forum, Vol. 32 (5), Wiley Online Library, 2013, pp. 113-123. doi:10.1111/cgf.12178.
4. S. Rusinkiewicz, M. Levoy, "Efficient variants of the icp algorithm", in: 3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on, IEEE, 2001, p. 145-152.
5. J. Salvi, C. Matabosch, D. Fofi, J. Forest, "A review of recent range image registration methods with accuracy evaluation", Image and Vision Computing 25 (5), Elsevier, 2007, p. 578-596.
6. G. Tam, Z.-Q. Cheng, Y.-K. Lai, F. Langbein, Y. Liu, D. Marshall, R. Martin, X.-F.

- Sun, P. Rosin, "Registration of 3d point clouds and meshes: A survey from rigid to nonrigid", *Visualization and Computer Graphics*, IEEE Transactions on 19 (7), 2013, p. 1199–1217. doi:10.1109/TVCG.2012.310.
7. D. Michael, H. Bischof. "3D segmentation by maximally stable volumes (MSVs)." ,18th International Conference on Pattern Recognition (ICPR'06). Vol. 1. IEEE, 2006. p.63-66
 8. J. Sun, M. Ovsjanikov, L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion", in: *Computer graphics forum*, Vol. 28, Wiley Online Library, 2009, pp. 1383–1392.
 9. Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3d object recognition", *Computer Vision Workshops (ICCV Workshops)*, 2009 IEEE 12th International Conference on, IEEE, 2009, p. 689–696.
 10. H. Pottmann, J. Wallner, Q.-X. Huang, Y.-L. Yang, "Integral invariants for robust geometry processing", *Computer Aided Geometric Design* 26 (1), Elsevier, 2009, p. 37–60.
 11. E. Boyer, A. M. Bronstein, M. M. Bronstein, B. Bustos, T. Darom, R. Horaud, I. Hotz, Y. Keller, J. Keustermans, A. Kovnatsky, et al., "Shrec 2011: robust feature detection and description benchmark", *3DOR2011-Eurographics Workshop on 3D Object Retrieval*, 2011, pp. 71–78.
 12. A. E. Johnson, M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes", *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 21 (5), 1999, p. 433–449.
 13. W.-l. Li, Z.-p. Yin, Y.-a. Huang, Y.-l. Xiong, "Automatic registration for 3d shapes using hybrid dimensionality-reduction shape descriptions", *Pattern Recognition* 44 (12), IEEE, 2011, pp. 2926–2943.
 14. C.-S. Chen, Y.-P. Hung, J.-B. Cheng, "Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images", *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 21 (11), IEEE, 1999, pp. 1229–1234.
 15. D. Aiger, N. J. Mitra, D. Cohen-Or, "4-points congruent sets for robust pairwise surface registration", in: *ACM Transactions on Graphics (TOG)*, Vol. 27, ACM, 2008, p. 85.
 16. N. Mellado, D. Aiger, N. J. Mitra, "Super 4pcs fast global pointcloud registration via smart indexing", in: *Computer Graphics Forum*, Vol. 33, Wiley Online Library, 2014, pp. 205–215.
 17. Y. Díez, F. Roure, X. Lladó, J. Salvi, "A qualitative review on 3d coarse registration methods", *ACM Computing Surveys (CSUR)* 47 (3), ACM, 2015, p.45.
 18. T. Masuda, K. Sakaue, N. Yokoya, "Registration and integration of multiple range images for 3-d model construction", in: *Pattern Recognition, 1996.*, Proceedings of the 13th International Conference on, Vol. 1, IEEE, 1996, pp. 879–883.
 19. N. Gelfand, L. Ikemoto, S. Rusinkiewicz, M. Levoy, "Geometrically stable sampling for the icp algorithm", in: *3-D Digital Imaging and Modeling*, Fourth International Conference on, IEEE, 2003, pp. 260–267.
 20. G. Godin, M. Rioux, R. Baribeau, "Three-dimensional registration using range and intensity information", in: *Photonics for Industrial Applications*, International Society for Optics and Photonics, 1994, pp. 279–290.
 21. H. Men, B. Gebre, K. Pochiraju, "Color point cloud registration with 4d icp algorithm", in: *Robotics and Automation (ICRA)*, 2011 IEEE International Conference on, IEEE, 2011, pp. 1511–1516.
 22. G. C. Sharp, S. W. Lee, D. K. Wehe, "Icp registration using invariant features", *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 24 (1), IEEE, 2002,

30 *Cejnog et. al.*

pp. 90–102.

23. D. Chetverikov, D. Svirko, D. Stepanov, P. Krsek, "The trimmed iterative closest point algorithm", in: *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, Vol. 3, IEEE, 2002, pp. 545–548.
24. J. Dong, Y. Peng, S. Ying, Z. Hu, "Lietricp: An improvement of trimmed iterative closest point algorithm", *Neurocomputing* 140, Elsevier, 2014, p. 67–76.
URL <http://dx.doi.org/10.1016/j.neucom.2014.03.035>
25. E. Trucco, A. Fusiello, V. Roberto, "Robust motion and correspondence of noisy 3-d point sets with missing data", *Pattern recognition letters* 20 (9), Elsevier, 1999, p. 889–898.
26. A. W. Fitzgibbon, "Robust registration of 2d and 3d point sets", *Image and Vision Computing* 21 (13), Elsevier, 2003, p. 1145–1153.
27. L. Maier-Hein, A. M. Franz, T. R. dos Santos, M. Schmidt, M. Fangerau, H. Meinzer, J. M. Fitzpatrick, "Convergent iterative closest-point algorithm to accomodate anisotropic and inhomogenous localization error", *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34 (8), IEEE, 2012, pp. 1520–1532.
28. A. Segal, D. Haehnel, S. Thrun, "Generalized-icp", in: *Robotics: Science and Systems*, Vol. 2, 2009.
29. J. Yang, H. Li, Y. Jia, "Go-icp: Solving 3d registration efficiently and globally optimally", *Computer Vision (ICCV), 2013 IEEE International Conference on*, IEEE, 2013, pp. 1457–1464.
30. J. Hermans, D. Smeets, D. Vandermeulen, P. Suetens, "Robust point set registration using em-icp with information-theoretically optimal outlier handling", *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, IEEE, 2011, pp. 2465–2472.
31. J. Servos, S. L. Waslander, "Multi channel generalized-icp", *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, IEEE, 2014, pp. 3644–3649.
32. L. Reyes, G. Medioni, E. Bayro, "Registration of 3d points using geometric algebra and tensor voting", *International Journal of Computer Vision* 75 (3), Springer, 2007, p. 351–369.
33. Albarelli, A., Rodol, E., Torsello, A., "Fast and accuratesurface alignment through an isometry-enforcing game", *Pattern Recognition*, 48(7), Elsevier, 2015, pp. 2209–2226.
34. G. Medioni, C.-K. Tang, M.-S. Lee, "Tensor voting: Theory and applications", *Proceedings of RFIA, Paris, France*, v 3, 2000.
35. M. Corsini, P. Cignoni, R. Scopigno, "Efficient and flexible sampling with blue noise properties of triangular meshes", *Visualization and Computer Graphics, IEEE Transactions on* 18 (6), IEEE, 2012, 914–924.
36. Horn, Berthold KP. "Closed-form solution of absolute orientation using unit quaternions." *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, on 4(4), 1987, pp. 629-642
37. L. W. X. Cejnog, "Rigid registration based on local geometric dissimilarity", Master's thesis, Universidade Federal de Juiz de Fora, 2015.

Photo and Bibliography

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



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



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