Rigid Registration of Noisy Point Sets Based on Tensor Structuring Elements

Fernando Akio de Araújo Yamada

JUIZ DE FORA

FEVEREIRO, 2014

Rigid Registration of Noisy Point Sets Based on Tensor Structuring Elements

Fernando Akio de Araújo Yamada

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Rodrigo Luis de Souza da Silva

Co-orientador: Marcelo Bernardes Vieira

JUIZ DE FORA

FEVEREIRO, 2014

Rigid Registration of Noisy Point Sets Based on Tensor Structuring Elements

Fernando Akio de Araújo Yamada

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Rodrigo Luis de Souza da Silva
Doutor em Engenharia Civil

Marcelo Bernardes Vieira
Doutor em Ciência da Computação

Saul de Castro Leite
Doutor em Modelagem Computacional

JUIZ DE FORA

07 DE FEVEREIRO, 2014

Resumo

O Iterative Closest Point (ICP) é um algoritmo tradicional de registro rígido. Este problema visa encontrar a transformação rígida que aplicada a uma nuvem de pontos $D \in \mathbf{R}^3$, a aproxima de outra malha $M \in \mathbf{R}^3$. Um problema comum do ICP é a presença de outliers nas nuvens, ocasionado por ruído. O método proposto, baseado em um processo de estimativa de normais através de elementos estruturantes tensoriais, é uma nova abordagem para lidar com outliers. Foi desenvolvido um fator comparativo de forma de tensores (CTSF, em inglês) que atua como fator de peso para a tradicional distância euclidiana, na etapa de casamento do ICP.

Palavras-chave: Registro rígido, Iterative Closest Point, elementos estruturantes tensoriais, estimativa de normal

Abstract

The Iterative Closest Point (ICP) is a traditional algorithm for rigid registration. In this problem, the goal is to find the rigid transformation that applied to a point cloud $D \in \mathbf{R}^3$, brings it closer to another point cloud $M \in \mathbf{R}^3$. A common problem within the ICP is the presence of outliers in the clouds, caused by noise. The proposed method, based on a normal estimation process through tensor structuring elements, is a novel approach to deal with outliers. A comparative tensor shape factor (CTSF) was developed, acting as a weighting factor to the traditional euclidian distance, in the ICP matching step.
Keywords: Rigid registration, Iterative Closest Point, tensor structuring elements, normal estimation

Acknowledgements

I would like to thank my parents, who always believed and trusted me; my aunts Cristina, Anna and Lucilia, my uncles Lauro and José Renato and their families for helping me whenever I needed; my friends in Barbacena, friends in GCG and GETComp for their friendship and support; and all those who helped me somehow.

A special thanks for Luciano Cejnog, Marcelo Bernardes and Rodrigo Luis. Without them this work would have not been possible.

Time can be the answer, take a chance, or lose it all
It's a simple mistake to make, to create love and to fall
So rise and be your master, cause you don't need to be a slave
Of memory ensnared in a web, in a cage
Anathema (A Simple Mistake)

Sumário

Lista de Figuras

Lista de Tabelas

Abbreviations List

DCC	Departamento de Ciência da Computação
UFJF	Universidade Federal de Juiz de Fora
ICP	Iterative Closest Point
CTSF	Comparative Tensor Shape Factor
RGB	Red, Green, Blue color model
HSL	Hue, Saturation, Lightness color space
CAD	Computer-Aided Design
SIFT	Scale-Invariant Feature Transform
RANSAC	Random Sample Consensus

1 Introduction

In computer vision a recurrent problem is surface registration. In this problem the aim is to find the spacial transformation that best aligns two or more point clouds into the same coordinate system. This process is essential to capture 3D geometry of objects, that can be used by CAD projects, movie animations, games or medical images, among other applications.

A point cloud is the given name to a set of point, where each element corresponds to a different point in space. It may contain information about its position, color or normal. These clouds can be generated synthetically or obtained from 3D Scanners, like the one presented by de Souza Filho *et al* [3]. The advent of low cost devices, such as the Microsoft Kinect or the Asus Xtion, enables a wider use of registration algorithms.

One of the most used algorithms to register point clouds is the Iterative Closest Point (ICP), presented by Besl and McKay [6]. It takes two point clouds, that could represent two distinct views of a scene or object and finds the rigid transformation that best approximates both clouds. Rusinkiewicz and Levoy [13] classify this algorithm in six stages where optimizations can be made:

1. **Selection** of the model points.
2. **Matching** the selected points to the other cloud.
3. **Weighting** the corresponding pairs.
4. **Rejecting** some pairs.
5. Assigning an **error metric**.
6. **Minimizing** the error metric.

The presence of outliers in the point cloud, due to low quality 3D scanning, could lead to a mismatch in the second step, which in turn can lead to a wrong transformation, producing bad results. In order to avoid this, a new weighting factor named CTSF

(Comparative Tensor Shape Factor) is proposed to compare two points in the matching step, reducing the influence of outliers.

For the proposed approach, first a local normal estimation is done for every point, considering its neighborhood. The result is a tensor for each point, containing encoded in its eigensystem a multivariate estimation of the normal. Next, in the ICP matching step, the CTSF compares the shape of this tensor, weighting the traditional euclidian distance.

The next chapters are organized as follows: chapter two presents the theoretical base of the Iterative Closest Point and normal estimation with tensor structuring elements. In chapter three, the Comparative Tensor Shape Factor is explained. Some results are shown in chapter four, and in the final chapter some conclusions are presented.

1.1 Problem Definition

Given two distinct point clouds $M, D \in \mathbf{R}^3$, find the rigid transformation $T(R, t)$ that applied to D, best aligns both clouds, where R is a rotation matrix and t is a translation vector.

1.2 Objectives

The main objective of this work is to improve the precision and robustness of the Iterative Closest Point (ICP) algorithm in the presence of strong non-structured noise, that is, a noise that does not form any surface. For this, a new approach to deal with outliers based on tensor structuring elements is proposed. The method computational efficiency will not be focused.

2 Theoretical Base

2.1 Iterative Closest Point

The Iterative Closest Point takes two point sets M and D, called Model and Data respectively, and finds the best rigid correspondence between them, i.e.,

$$\min_{R,t} \sum_{i=1}^{n_D} \|RD_i + t - M_i\|^2 , \tag{2.1}$$

where n_D is the number of point in D, R is the rotation matrix and t is the translation vector that applied to D brings it closer to M. It has two major steps: matching of the points, often called nearest neighbor search, and transformation estimative.

In the matching step, for each point $M_i \in M$, the closest point $D_i \in D$ is found, composing the set of closest points C. This step is usually the slowest part of the algorithm, and a K-D Tree is a common data structure to accelerate it. Outliers can lead to a mismatch, building an erroneous set C. Thus it is necessary to develop outlier detection techniques.

The transformation estimative step finds the covariance between the sets C and M, estimating a rigid transformation matrix to be applied on every point of the data set. In the original ICP, Besl and McKay [6] used a quaternion-based approach to find the transformation, but there are several alternatives that can be used, such as the presented by Chen and Medioni [1].

These two steps are performed until a stopping criterion is satisfied.

2.1.1 Point-to-Plane

The point-to-plane technique by Chen and Medioni [1], implemented in many variants of the ICP, allows greater tangencial movements, converging faster to a local minimum. This method is iterative and fits as an optimization on the matching step of the ICP, replacing the nearest neighbour search.

Let $p \in M$, be a point of the model set, where the normals are defined for each point. For each iteration k, and starting $r_0 = p$, find the closest point $q_k \in D$, another point set where the normals are also defined. The intersection between a line in the direction of the normal of r_k and the tangent plane defined by the normal of q_k defines a point r_{k+1}. Repeat the iterative step until $\|r_k - r_{k+1}\| > \epsilon$ or $k < n$, for a given ϵ. The closest point set C is build from q_k.

Chen and Medioni also presented a point-to-plane minimization function as an extension to the point-to-point function, in equation 2.1, but considering the normal n_i of each point in the data set. These normals can be computed based on the four nearest neighbors in the range grid [13], or using a more sophisticated method like the presented on section 2.2. It can be written as

$$E = \sum_{i=1}^{n_D} \|(RD_i + t - M_i) \cdot n_i\|^2 . \tag{2.2}$$

The transformation estimative is done considering a linearization of the rotation matrix, assuming a small angular displacement ($\cos\theta = 1$ and $\sin\theta = \theta$). The error introduced here tends to be small in late iterations of the ICP, since both sets should be almost alligned. The linearized rotation matrix is:

$$R \approx \begin{pmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{pmatrix}, \tag{2.3}$$

with α, β, γ as rotations around the $\hat{x}$, $\hat{y}$ and $\hat{z}$ axis, respectively.

Substituting 2.3 into 2.2, it can be rewritten as:

$$\begin{aligned} E = \sum_{i=1}^{n_D} [(D_i - M_i) \cdot n_i \quad &+ \quad t \cdot n_i + \alpha\,(D_{i,y}n_{i,z} - D_{i,z}n_{i,y}) + \beta\,(D_{i,z}n_{i,x} - D_{i,x}n_{i,z}) \\ &+ \quad \gamma\,(D_{i,x}n_{i,y} - D_{i,y}n_{i,x})]^2. \end{aligned}$$

Defining:

$$c_i = D_i \times n_i$$

and

$$r = \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix},$$

the equation 2.4 becames:

$$E = \sum_{i=1}^{n_D} [(D_i - M_i) \cdot c_i + t \cdot n_i + r \cdot c_i]^2.$$

Minimizing this function with respect to α, β, γ, t_x, t_y and t_z, the following linear system can be set:

$$\sum_{i=1}^{n_D} \begin{pmatrix} c_{i,x}c_{i,x} & c_{i,x}c_{i,y} & c_{i,x}c_{i,z} & c_{i,x}n_{i,x} & c_{i,x}n_{i,y} & c_{i,x}n_{i,z} \\ c_{i,y}c_{i,x} & c_{i,y}c_{i,y} & c_{i,y}c_{i,z} & c_{i,y}n_{i,x} & c_{i,y}n_{i,y} & c_{i,y}n_{i,z} \\ c_{i,z}c_{i,x} & c_{i,z}c_{i,y} & c_{i,z}c_{i,z} & c_{i,z}n_{i,x} & c_{i,z}n_{i,y} & c_{i,z}n_{i,z} \\ n_{i,x}c_{i,x} & n_{i,x}c_{i,y} & n_{i,x}c_{i,z} & n_{i,x}n_{i,x} & n_{i,x}n_{i,y} & n_{i,x}n_{i,z} \\ n_{i,y}c_{i,x} & n_{i,y}c_{i,y} & n_{i,y}c_{i,z} & n_{i,y}n_{i,x} & n_{i,y}n_{i,y} & n_{i,y}n_{i,z} \\ n_{i,z}c_{i,x} & n_{i,z}c_{i,y} & n_{i,z}c_{i,z} & n_{i,z}n_{i,x} & n_{i,z}n_{i,y} & n_{i,z}n_{i,z} \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ t_x \\ t_y \\ t_z \end{pmatrix} = -\sum_{i=1}^{n_D} \begin{pmatrix} c_{i,x}(D_i - M_i) \cdot n_i \\ c_{i,y}(D_i - M_i) \cdot n_i \\ c_{i,z}(D_i - M_i) \cdot n_i \\ n_{i,x}(D_i - M_i) \cdot n_i \\ n_{i,y}(D_i - M_i) \cdot n_i \\ n_{i,z}(D_i - M_i) \cdot n_i \end{pmatrix}.$$

This linear system is in the form $Ax = b$, solving it will give the values to build the final transformation matrix:

$$T = \begin{pmatrix} 1 & -\gamma & \beta & t_x \\ \gamma & 1 & -\alpha & t_y \\ \beta & \alpha & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

2.1.2 Related Works

After the introduction of the Iterative Closest Point by Besl and McKay [6], many improvements were made focusing on robustness, speed or both. The survey from Tam *et al.* [15] presents several recent proposals to rigid and non-rigid registration, showing that both problems are still open, although non-rigid registration is still taking its initial steps. Their work classifies method by its optimization strategy, being stochastic, local deterministic or global deterministic and by the constraints set, that could be transformation-induced, features, saliency, regularization and search. The classical ICP and the version presented in this work fits in the transformation-induced constraint in their classification, because of its closest point criterion. It is also local deterministic, since it finds the best transformation locally at each step.

The K-D Tree [5] is a data structure often implemented to accelerate the closest point search step [14]. In the context of the ICP, the tree usually uses the 3D cartesian space. Hao Men *et al.* [10] uses a four-dimensional K-D Tree, setting the fourth axis as the hue component, from a HSL color model. In the work of Henry *et al.* [8], color is used aside feature detection (SIFT) and RANSAC to optimize the allignment when geometric information is not enough. Druon *et al.* [7] also uses hue to improve the registration quality, subdividing the cloud in seven basic colors and performing individual matches for each subcloud, before estimate a final transformation. The drawbacks of using color are the need of a reasonably well lit capture environment of the point clouds, since color detection is sensitive to brightness, and innacurate on reflective surfaces. The proposed method does not use color and, therefore, is not subject to these problems.

Since outliers can mislead the algorithm in the matching step, many methods to identify them have been proposed. In the work of Phillips *et al.* [11], a probabilistic method to identify inliers is used to modify the distance function. Another probabilistic method is shown by Hermans *et al.* [9], where Gaussian Mixture Models are used to model the point cloud and an expectation-maximization process is adapted into the ICP to handle outliers. KinectFusion [2] is a parallel implementation of the traditional ICP to achieve real-time registration, identifying outliers through a segmentation process and eliminating the residuals using raycasting. Sparsity-induced norms are used by Bouaziz *et al.* [4] to modify the penalty function applied, reducing the influence of outliers on the transformation estimative.

Reyes *et al.* [12] estimate the rigid transformation using geometric algebra, modeling the problem as finding a 3D plane in a joint space which represents the affine motion. A process named tensor voting is used to find this plane. In this work, the method used to estimate normals is similar to this process, however the structuring element used is anisotropic and the gradient of normals coincide with the tensor orientation, producing smoother results [16].

2.2 Normal estimation with tensor structuring elements

This method is strongly based on the work of Vieira *et al* [16] and is divided in two steps: coarse and fine. The first step provides an initial estimative, based only on the nearby neighborhood geometry. This estimative is used as a starting point to the second step, where another structuring element is applied, refining the estimative. The final result is a tensor for each point, containing encoded in its eigensystem a multivariate estimation of the normal. These tensors are used by the CTSF during the ICP matching step in the proposed approach.

In the coarse step a tensor $\mathbf{T_p}$ is build for each point applying a radial tensor structuring element. This way, for each $p, q \in M$ and $p \neq q$:

$$\mathbf{T_p} = \sum_{q \in M} e^{\frac{-||\vec{pq}||^2}{\sigma^2}} \cdot \hat{pq} \cdot \hat{pq}^T,$$

where $\hat{pq}$ is the normalized vector $\vec{pq}$ and σ is a parameter.

The initial normal estimative is the eigenvector associated to the less significant eigenvalue of $\mathbf{T_p}$. A confidence value, used to measure how good is the estimative, can be the planar anisotropy coefficient c_p, defined as:

$$c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3}.$$

Where $\lambda_1, \lambda_2, \lambda_3$ are the eigenvalues. When $c_p \gg c_l \approx c_s$, the tensor has an indecibility on the third main direction, indicating a bigger chance of belonging to a planar surface.

In the second step another tensor, $\mathbf{S_p}$, is built for each point. The tensor will store the influence exerted by the neighborhood of p, based on the distance over an elliptic trajectory between p and its neighbors.

First, the structuring element must be alligned with the estimated normal. For that, the applied rotation matrix is given by:

$$R_p = \begin{bmatrix} e_{1x} & e_{1y} & e_{1z} \\ e_{2x} & e_{2y} & e_{2z} \\ e_{3x} & e_{3y} & e_{3z} \end{bmatrix},$$

where e_1, e_2, e_3 are the normalized eigenvectors of $\mathbf{T_p}$. Then, for each $p, q \in M$ and $p \neq q$, the vector $\vec{pq}'$ is calculated as follows:

$$\vec{pq}' = R_p \cdot \vec{pq}.$$

This vector expressed in spherical coordinates is:

$$\begin{cases} \rho = \sqrt{pq'^2_x + pq'^2_y + pq'^2_z}, \\ \theta = \tan^{-1} \frac{pq'_y}{pq'_x}, \\ \phi = \tan^{-1} \frac{pq'^2_z}{\sqrt{pq'^2_x + pq'^2_y}}. \end{cases}$$

Let a family of ellipsoids be defined as:

$$\frac{x^2}{t_x^2 k^2} + \frac{(-t_y + \frac{y}{k}))}{t_y^2} = 1.$$

The curvature of this family can be controlled by the ratio d:

$$d = \frac{2kt_y}{2kt_x} = \frac{t_y}{t_x}.$$

Then, let E be an ellipsoid centered over the $\hat{y}'$-axis and tangent to the $\hat{x}'$-axis. The angle β between the plane $\hat{x}'\hat{z}'$ and the plane tangent to q on the elliptic surface can be calculated:

$$\beta = \tan^{-1} \frac{2d^2 \tan\theta}{d^2 - \tan^2\theta}$$

Back to cartesian coordinates, replacing β for ϕ, a vector orthogonal to E on q'

can be obtained:

$$\vec{v_{pq'}} = \cos\theta \cdot \cos\beta \cdot \hat{x} + \sin\theta \cdot \cos\beta \cdot \hat{y} + \sin\beta \cdot \hat{z}.$$

The distance between p and q over the elliptic trajectory is:

$$d(pq') = \rho \cos\phi (1 + (2 - \frac{1}{d^2}) \tan^2\phi)^{\frac{d^2}{2d^2-1}}.$$

For the given σ, the influence force that q exerts on p is:

$$f(pq') = e^{\frac{-d(pq')^2}{\sigma^2}}.$$

Finally, the resulting tensor $\mathbf{S_p}$ is defined as:

$$\mathbf{S_p} = \sum_{\substack{q \in M \\ \phi_{\vec{pq'}} \leq \frac{\pi}{4}}} f(pq') \cdot \vec{v_{pq'}} \cdot \vec{v_{pq'}}^T.$$

The restriction on ϕ constrains the influence of q misaligned to the tangent plane defined by the normal of p. A higher value of ϕ produces smoother surfaces, while smaller values allows more details at cost of robustness to noise [16].

Like the previous tensor, the eigenvector associated to the less significant eigenvalue is the normal estimated, and the used confidence factor is the planar anisotropy coefficient c_p.

3 Proposed Method

Given two points belonging to different point sets, when their neighborhood have the same geometric arrangement, it is very likely that they are the same point. This way, the influence exerted by its neighborhood is the same in both sets, so that the tensors obtained in the normal estimation step are equal. If one of the point sets is denser, the point will receive more influence from the neighborhood, producing a tensor with greater magnitude, but its shape will still be the same. To avoid the influence of the magnitude, both tensors must be normalized during the process.

In order to be able to compare two tensors, and determine how compatible they are, a Comparative Tensor Shape Factor (CTSF) is created. Representing the tensor as an ellipsoid, its shape is defined directly by its eigenvalues. This way, when a tensor has a well defined direction its shape is a stretched rod, when there is an indecibility on the two main directions its shape is like a disc, and when there is an indecibility on the three main directions its shape is of a sphere. Figure 3.1 shows how the shape is affected by the eigenvalues.

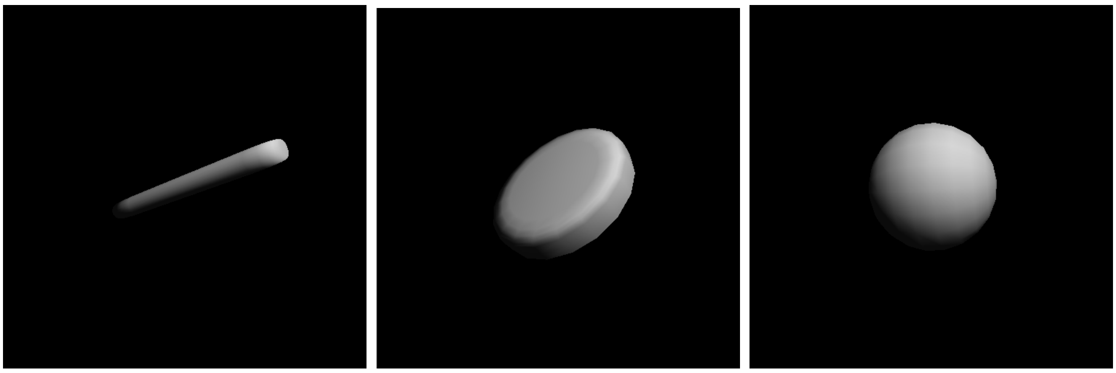


Figura 3.1: Tensor shape according to how many indecibility directions it have.

Two tensors have the same shape when they have the same proportion between their eigenvalues. Hence, the CTSF is defined as:

$$CTSF\left(\hat{S}_1, \hat{S}_2\right) = \sum_k \left(\lambda_k^{\hat{S}_1} - \lambda_k^{\hat{S}_2}\right),$$

with $\lambda_k^{\hat{S}_m}$ as the k-th greatest eigenvalue of $\hat{S}_m$. The greater the CTSF, more dissimilar the tensors are, i.e., more different are their shapes. In figure 3.2, some examples are shown of when two tensors have low or high values of CTSF. Since eigenvalues are invariant to rigid transformations, this factor is suitable to rigid registration.

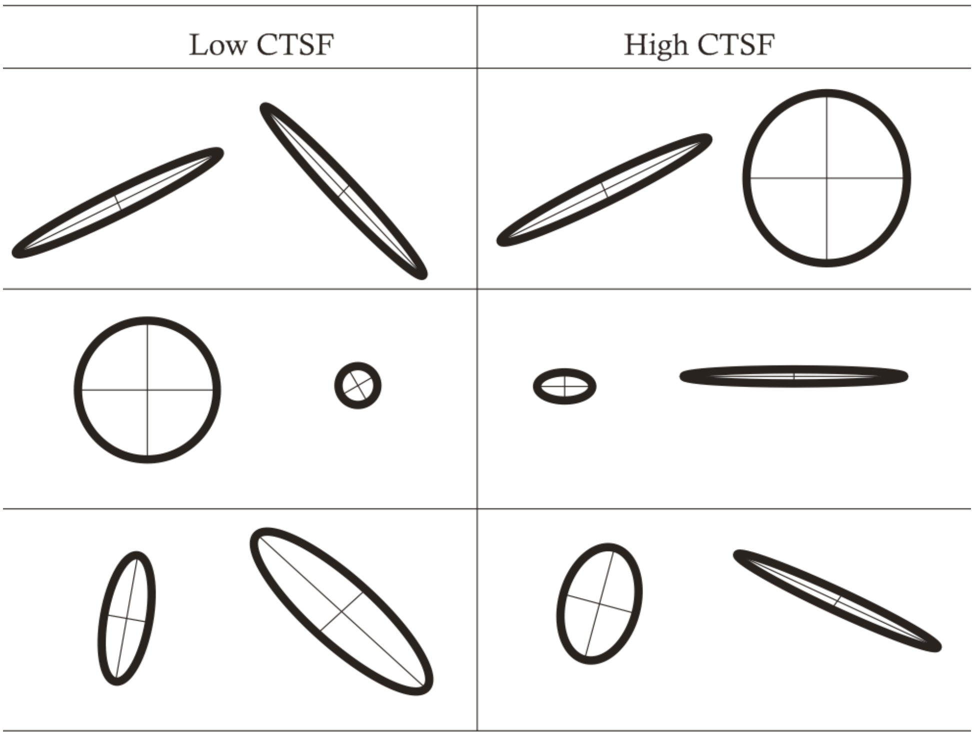


Figura 3.2: Examples of how the CTSF is affected by the geometry of planar tensors. Note that the CTSF is invariant to the magnitude of tensors, due to the normalization, and to their orientation.

During the search for the nearest neighbor in the ICP matching step, the algorithm must find the closest point using an euclidian distance. The CTSF is used in the exponential form to weight this distance. Thus, for given two points $m \in M$ and $d \in D$, instead of only measuring their distance, the ICP must also consider how similar they are:

$$e^{\frac{-CTSF(\hat{S}_m, \hat{S}_d)^2}{\sigma^2}} \cdot ||\vec{md}||.$$

The exponential was chosen so that when two tensors have the exactly same shape, only their euclidian distance is considered, since the weighting term becames 1 in this case. Otherwise, when two tensors differs their shape too much, the weighting factor becames bigger than 1, lowering the chance to match. The σ used in this case is the same as the used in the normal estimative step.

4 Results

4.1 Normal Estimative

The Figures 4.1, 4.2, 4.3 show how reliable are the normals estimated using the tensor structuring elements. The models were drawn using the ellipsoid representation of the tensors found in the normal estimative step. Red tensors represents a high value of c_p, that is, a more reliable estimative; while blue tensors, in the other hand, represent an uncertainty.

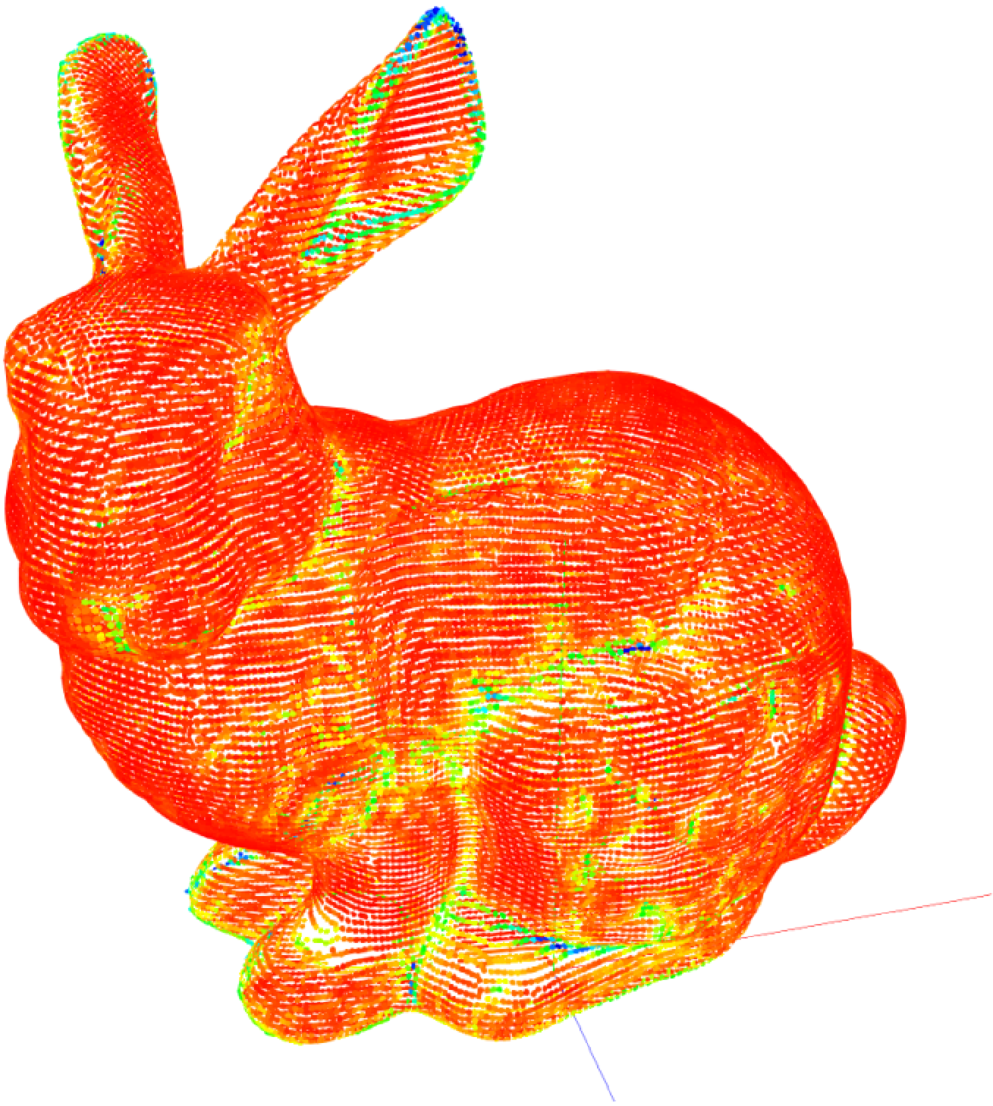


Figura 4.1: The bunny set has very reliable normals along its body. Except the edge of the ears, a region with high curvature, which is harder to estimate normals.

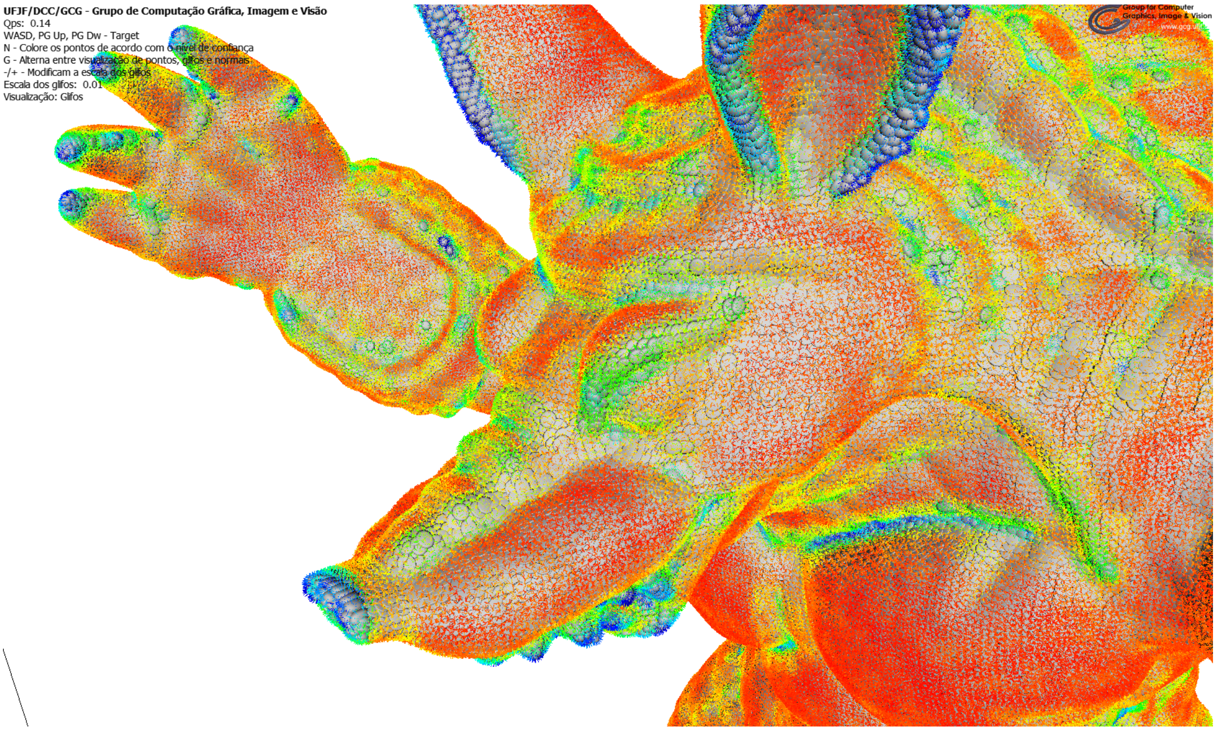


Figura 4.2: The chest, muzzle, palm of hand and cheeks of the armadillo are regions with very well defined normals, opposed to fingertips, teeth, nose and the edge of the ears, regions with high curvature.

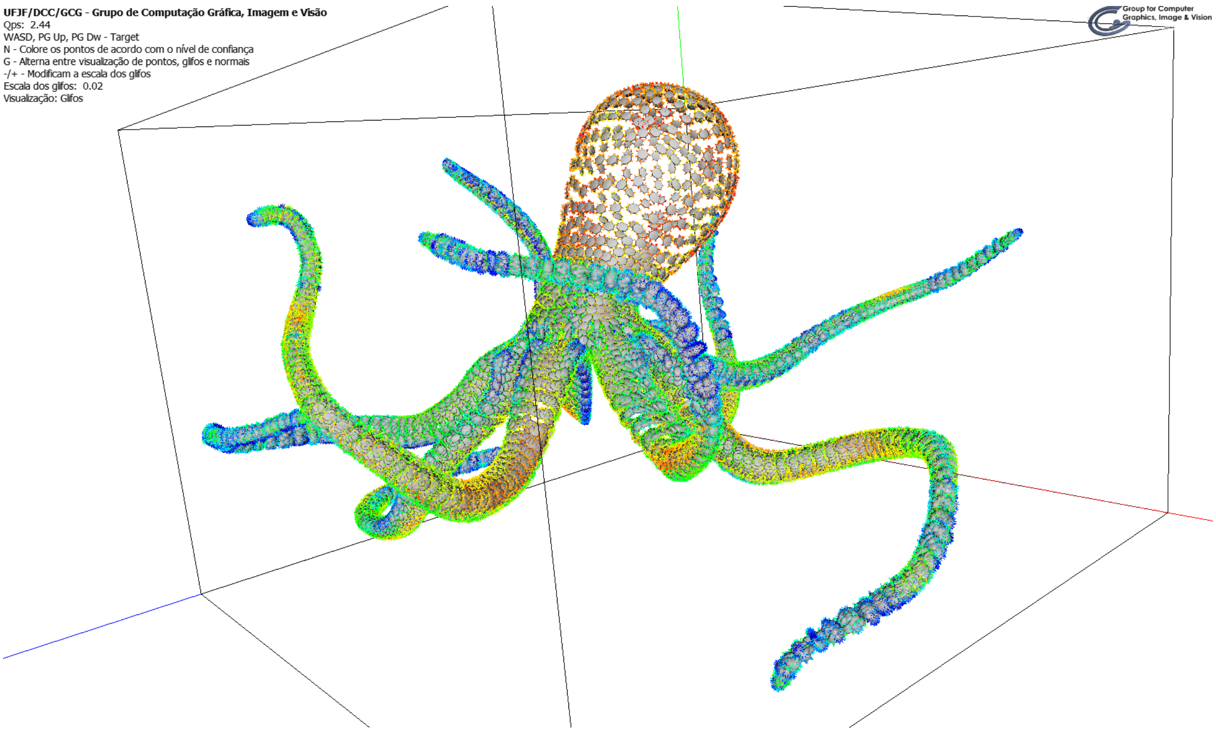


Figura 4.3: In the octopus set only the head and small patches in tentacles have normals with high confidence. The rest of the tentacles have medium to low confidence.

4.2 Iterative Closest Point + CTSF

The experiments were made on the point sets *Bunny* (from the Stanford 3D Scanning Repository, graphics.stanford.edu/data/3Dscanrep, containing 8171 points), *Egea* and *Octopus* (from the Madras Repository, www-rech.telecom-lille1.eu:8080/3dsegbenchmark/dataset.html, containing 8268 and 16554 points respectively). To show how the ICP behaves, a synthetical rigid transformation was applied to the data set, while the model set remained unchanged. The transformation is composed by a translation of 0.5 in each axis, and a rotation of 20° around the $\hat{z}$ axis. Since the ICP needs a small displacement between the point sets, this transformation represent a feasible scenario.

In the proposed method the only variable parameter is σ, and the test values were 0.01, 0.1, 1.0, 10.0.

Non-strucured noise was applied to both point sets to show how the method deals with it. The amount of noise added was proportional to 0.5, 1 and 2 times the number of points, generated randomly following a uniform distribution, inside a bounding box two times larger than the original. The original ICP used was the one based on quaternions, proposed by Besl and McKay [6] and accelerated by a K-D Tree.

σ	Original	Egea +50%	Egea +100%	Egea +200%
0.01	0.611223	0.589953	0.663178	0.765912
0.10	0.614304	0.528759	0.0658112	0.732408
1.00	0.233932	0.025418	0.026132	0.025831
10.00	0.232850	0.024968	0.026178	0.026660
Original ICP	0.067624	0.032374	0.062652	0.035598

Tabela 4.1: Root mean square errors for the *Egea* point set, varying the amount of noise and the parameter σ. The gray values are the best results for each model.

σ	Original	Octopus +50%	Octopus +100%	Octopus +200%
0.01	0.373363	0.539407	0.614426	0.612511
0.10	0.297609	0.377939	0.444942	0.475045
1.00	0.004846	0.041921	0.018494	0.019082
10.00	0.004166	0.042753	0.042753	0.019252
Original ICP	0.085117	0.035529	0.052943	0.035329

Tabela 4.2: Root mean square errors for the *Octopus* point set, varying the amount of noise and the parameter σ. The gray values are the best results for each model.

Table 4.1 and 4.2 shows that the CTSF improved the Original ICP when a proper value for σ is chosen, except for only one test case for each point set. The best value for sigma was between 1 and 10, when the CTSF was used. A much lower σ implies that the structuring element is unable to gather enough information on the neighborhood of the points, producing poor tensor at the normal estimative step. These bad tensors tend to be confused with outliers, causing a mismatch at the ICP matching step. On the other hand, a much bigger value for σ does not produce better results, keeping the RMS error close to the one when $\sigma = 10$.

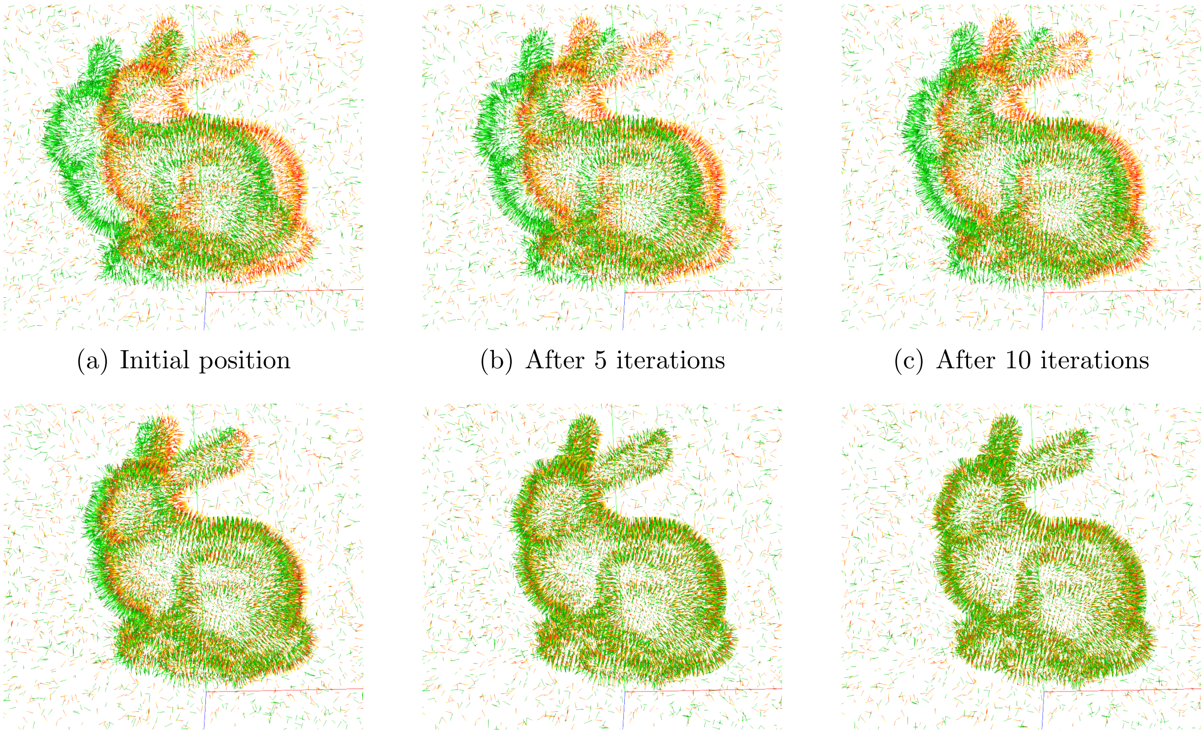
Another synthetical test was performed. A set of 25 random transformations was applied to the data set, while the model set remained unchanged, just like the previous experiment. The transformations were composed by a translation of 0.5 in each axis, and a rotation of 20º, 40º and 60º around an arbitrary axis, with the following values for σ: 0.01, 0.1, 1.0, 10.0. The Table 4.3 shows the results for 20°, 40° and 60°. In this table only the best average value for σ is shown when the ICP + CTSF is used.

Bunny	**ICP + CTSF**			**Original ICP**	
20°	**Sigma**	**Average**	**Std Deviation**	**Average**	**Std Deviation**
Original	10	0,01226	0,007578772	0,20524	0,240917012
50%	1	0,009903	0,004704758	0,241066	0,154638192
100%	10	0,006947	0,002458747	0,053733	0,048210172
Bunny	**ICP + CTSF**			**Original ICP**	
40°	**Sigma**	**Average**	**Std Deviation**	**Average**	**Std Deviation**
Original	10	0,02025	0,003979782	0,179869	0,23438386
50%	10	0,012208	0,001751649	0,260375	0,125582949
100%	10	0,01117	0,00061695	0,102806	0,131441182
Bunny	**ICP + CTSF**			**Original ICP**	
60°	**Sigma**	**Average**	**Std Deviation**	**Average**	**Std Deviation**
Original	10	0,017422	0,003615576	0,242524	0,22489159
50%	10	0,013184	0,000936384	0,354515	0,128023817
100%	10	0,01139	0,001629041	0,087954	0,075503955

Tabela 4.3: Average RMS error and standard deviation for the *Bunny* point set, with a rotation of 20°, 40° and 60° in an arbitrary axis and a translation of 0.5 in each axis, varying the amount of noise and comparing with the Original ICP.

Table 4.3 shows that the predominant value for σ in the best case is again close to 10. In every test cases the presented method performed better than the Original ICP. The presence of noise demonstrates even more the effectiveness of the method. The presented method proved to be much more stable than the Original ICP, given the low variation of the standard deviations. These results, however, are not absolute, since the transformations used were different. They only indicate a trend.

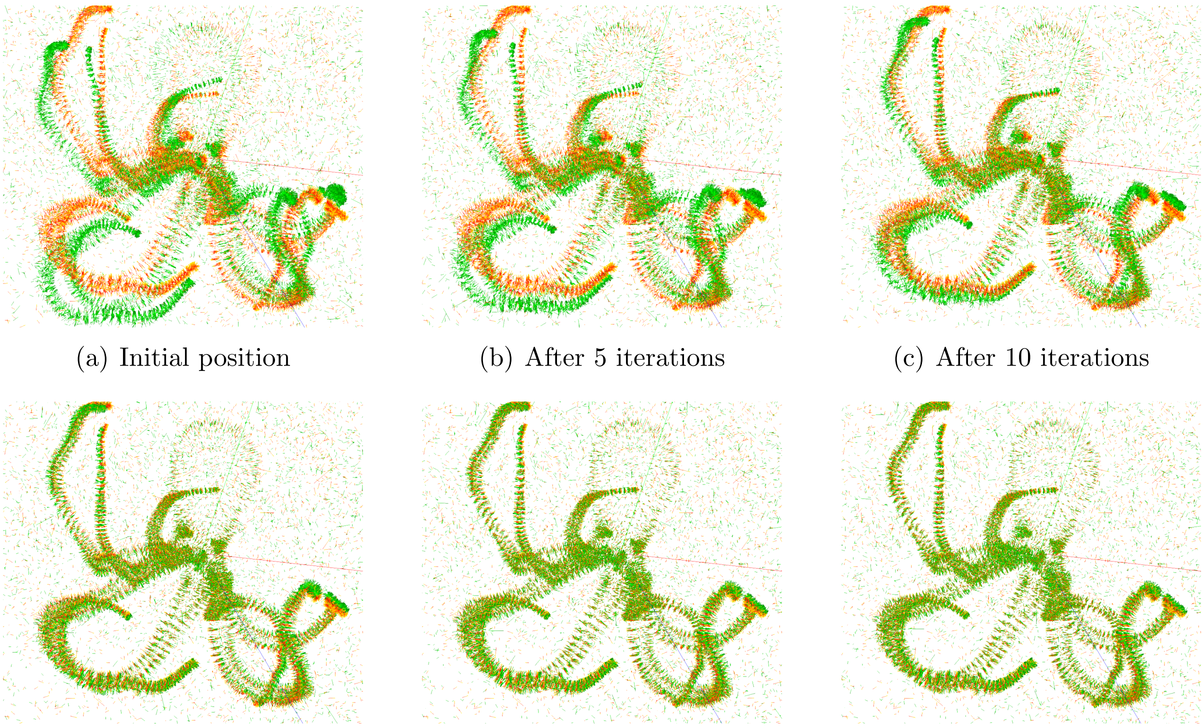
The ICP + CTSF usually takes more iterations than the original ICP, in some cases more than 150 iterations, an unusual behavior for the ICP. Due to the quadratic nature of the normal estimative step, the whole process is slow, a tradeoff chosen between precision over speed. For this reason, a speed comparison is not interesting. To reach a feasible result, an inverted list of neighbors was used.



(a) Initial position (b) After 5 iterations (c) After 10 iterations

(d) After 20 iterations (e) After 30 iterations (f) Final result after 37 iterations

Figura 4.4: Method convergence sequence for the *Bunny* dataset, considering $\sigma = 0.5$, and an amount of 50% of noise.



(a) Initial position (b) After 5 iterations (c) After 10 iterations

(d) After 20 iterations (e) After 30 iterations (f) Final result after 39 iterations

Figura 4.5: Method convergence sequence for the *Octopus* dataset, considering $\sigma = 1.0$, and an amount of 50% of noise.

5 Conclusion

As stated in the related works, rigid registration is still an open problem. The proposed method using a novel approach, the use of tensor structuring elements in the ICP through the CTSF, in the presence of strong non-structured noise, is the major contribution of this work.

The first step is the estimative of the normals for each point in both clouds, based on how likely its neighbors compose a surface, resulting in a tensor for each point. Then, in the ICP matching step, a Comparative Tensor Shape Factor (CTSF) is used in order to compare two tensors, weighting the traditional euclidian distance, mitigating the influence of outliers in the transformation estimative step. It is important to note that any point is discarded in the process.

The only variable parameter is σ, responsible for controlling the influence range of the structuring element and the CTSF. In most of the tests performed the ideal range is between 1 and 10, but this is highly sensitive to the point sets used, even with all sets scaled to the unit bounding box.

As a future work, an adaptive σ might be investigated, eliminating the need to calibrate the parameter for each new input point set. Different kinds of noise can be explored, like additive noise for example. The CTSF itself is way to compare the shape of any given two tensors, and can be used whenever is interesting to compare tensors, expanding its usage beyond the ICP matching step.

This work was submitted to the International Conference on Pattern Recognition (ICPR) 2014, on 21/12/2013.

Referências Bibliográficas

[1] Yang, C.; Medioni, G. Object modelling by registration of multiple range images. **Image and vision computing**, v.10, n.3, p. 145–155, 1992.

[2] Izadi, S.; Kim, D.; Hilliges, O.; Molyneaux, D.; Newcombe, R.; Kohli, P.; Shotton, J.; Hodges, S.; Freeman, D.; Davison, A. ; others. **Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera**. In: Proceedings of the 24th annual ACM symposium on User interface software and technology, p. 559–568. ACM, 2011.

[3] de Souza Filho, J. L.; Silva, R. C.; Sad, D. O.; Dembogurski, R.; Vieira, M. B.; de Oliveira Dantas, S. ; Silva, R. **Analysis of a high definition camera-projector video system for geometry reconstruction**. In: Proceedings of the 12th international conference on Computational Science and Its Applications-Volume Part I, p. 228–239. Springer-Verlag, 2012.

[4] Bouaziz, S.; Tagliasacchi, A. ; Pauly, M. Sparse iterative closest point. **Computer Graphics Forum**, v.32, n.5, p. 113–123, 2013.

[5] Bentley, J. L. Multidimensional binary search trees used for associative searching. **Communications of the ACM**, v.18, n.9, p. 509–517, 1975.

[6] Besl, P. J.; McKay, N. D. **Method for registration of 3-d shapes**. In: Robotics-DL tentative, p. 586–606. International Society for Optics and Photonics, 1992.

[7] Druon, S.; Aldon, M.-J. ; Crosnier, A. **Color constrained icp for registration of large unstructured 3d color data sets**. In: Information Acquisition, 2006 IEEE International Conference on, p. 249–255. IEEE, 2006.

[8] Henry, P.; Krainin, M.; Herbst, E.; Ren, X. ; Fox, D. **Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments**. In: the 12th International Symposium on Experimental Robotics (ISER), volume 20, p. 22–25, 2010.

[9] Hermans, J.; Smeets, D.; Vandermeulen, D. ; Suetens, P. **Robust point set registration using em-icp with information-theoretically optimal outlier handling**. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, p. 2465–2472. IEEE, 2011.

[10] Men, H.; Gebre, B. ; Pochiraju, K. **Color point cloud registration with 4d icp algorithm**. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on, p. 1511–1516. IEEE, 2011.

[11] Phillips, J. M.; Liu, R. ; Tomasi, C. **Outlier robust icp for minimizing fractional rmsd**. In: 3-D Digital Imaging and Modeling, 2007. 3DIM'07. Sixth International Conference on, p. 427–434. IEEE, 2007.

[12] Reyes, L.; Medioni, G. ; Bayro, E. Registration of 3d points using geometric algebra and tensor voting. **International Journal of Computer Vision**, v.75, n.3, p. 351–369, 2007.

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

[14] Simon, D. A. **Fast and accurate shape-based registration**. 1996. Tese de Doutorado - Carnegie Mellon University.

[15] Tam, G.; Cheng, Z.-Q.; Lai, Y.-K.; Langbein, F.; Liu, Y.; Marshall, D.; Martin, R.; Sun, X.-F. ; Rosin, P. Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. **Visualization and Computer Graphics, IEEE Transactions on**, v.19, n.7, p. 1199–1217, 2013.

[16] Vieira, M. B.; Martins, P.; Araujo, A.; Cord, M. ; Philipp-Foliguet, S. **Smooth surface reconstruction using tensor fields as structuring elements**. In: Computer Graphics Forum, volume 23, p. 813–823. Wiley Online Library, 2004.