University Of California - Los Angeles

Computer Science Department


TR070005

# ICPSAC: A New Robust and Accurate Registration Algorithm

Luigi Ricciato – Stefano Soatto

March 22, 2007

## Acknowledgments

I wish to express all my gratitude to all those persons who have given me a contribute for obtaining success with this thesis. First of all, I wish to acknowledge the Prof. Stefano Soatto who, by hosting me in his Vision Laboratory at the UCLA and by letting me work with wonderful people, has given me the great opportunity of learning and improving my skills. Also, I have to acknowledge the Prof. Ruggero Frezza without whom I would have never had a similar extremely good experience. Then, I wish to acknowledge Daniele Fontanelli for his precious suggestions. Finally, I would want to thank everybody who, even if not directly relating with my work, have given a contribute for a happy and pleasant stay in Los Angeles: Alessandro, Andrea, Gregorio, Veronica, Alexandro, Giuseppe and Matteo.

"This work was supported by AFOSR under contract FA9550-06-1-0138."

# Contents

# Notation

- $\mathcal{M}$: model points set
- $\mathcal{S}$: scene points set
- $\mathbf{m}_i$: $i$-th model's point
- $\mathbf{s}_i$: $i$-th scene's point
- $R^T$: R transposed
- $R^{-1}$: R inverse
- $SO(3)$: Special Orthogonal Matrices Set,

$$SO(3) := \{R \in \mathbb{R}^{3\times 3} \mid R^T R = I,\, det(R) = +1\}$$

- $\vec{v}$: vector
- $< M >$: model reference frame
- $< S >_k$: scene reference frame at the step $k$
- $< L^i >$: reference frame of the $i$-th lidar scan
- $g_{sm}$: transformation from the *model* to the *scene* reference frame
- $\mathbf{a}_{sm}^k$: parameter's vector at the time $k$ of the transformation $g_{sm}$
- $R_{sm}$: rotation matrix from the $< M >$ to the $< S >$ reference frame
- $T_{sm}$: translation vector from the $< M >$ to the $< S >$ reference frame
- $\mathbf{x}_s^{\mathbf{q}}$: coordinate of the point $\mathbf{q}$ expressed with respect to $< S >$
- $\mathbf{x}_m^{\mathbf{q}}$: coordinate of the point $\mathbf{q}$ expressed with respect to $< M >$
- $\|\cdot\|$: Euclidean Norm
- $\langle\cdot,\cdot\rangle$: Inner Product

# Summary

With this thesis, a strategy for building reconstruction relying on LIDAR data only is presented. It provides a comprehensive introduction into the field of reconstrunction from LIDAR data, for both indoor and outdoor mapping. We describe and compare various already existing techniques and propose a largely improved algorithm based on Point Set Registration by using a RANSAC-based ICP (ICPSAC) approach that makes it a robust and accurate point set registration algorithm in presence of noise and large amount of outliers.

Simulated and experimental results show that our algorithm achieves very good performance in terms of both robustness and accuracy in a relatively short time.

# Chapter 1

# Introduction

## 1.1 Abstract

The measured data acquired by sampling the same scene at different time instants and from different perspectives, is expressed in different coordinate frames. Registration is the process of coherently expressing the different sets of data into one coordinate system. Our algorithm must face a problem which has not been satisfactorily solved yet: finding an accurate transformation between two point sets (measurements obtained at different time instants) against quite large rotation and translation, noise and strong outliers presence. We will show how the algorithm is stable even in difficult work conditions. In fact it can properly deal with occlusions (i.e., objects that partially appear in one measurement) and it has been successfully tried with low-overlapping scenes.

## 1.2 Related Work

Registration of point sets is an important component in many computer vision problems such as motion tracking, face recognition, imaged guided surgery and robot navigation. For these reasons, extensive studies on the point set registration and related problems can be found in a rich literature covering those teoretical and practical issues.

A very common class of approaches is based on the *Iterative Closest Point* (ICP) algorithm proposed by Besl and McKay [3]. This is a standard solution to the alignment problem. Nonetheless, it has its own limitations, like the local convergence problem which requires sufficient overlap between the datasets and a close inizialization. Also, a naive implementation of ICP is known to be prone to outliers.

Therefore, since the introduction of ICP, many variants have been intro-

duced on the basic ICP concept, seeking to improve robustness, convergence and precision affecting all phases of the algorithm from the selection and matching of points to the minimization strategy.

For instance, Fitzgibbon [4] presents an attempt to directly, rather than iteratively, minimize the registration error using a general-purpose nonlinear optimization, that is the Levenberg-Marquardt algorithm. Chetverikov et al. [5] propose a robustified extension based on the consistent use of the Least Trimmed Squares approach in all phases of the operation, but needs the overlap between the two point sets has to exceed 50%. However the major drawback of these approaches is the necessity of a roughly preregistration between the two data sets or at least a good guess on the initial transformation to guarantee the correct solution can be found. Yet, they are not robust to large outliers amount.

Another interesting class of solutions involves methods that align two point sets without establishing the explicit point correspondence, and thus achieve a good robustness to the missing correspondences and outliers. The idea is to model each of the two point sets by a kernel density function and then quantify the similarity between them using a certain distance measure between probability density functions, e.g. the *Kullback-Leibler* divergence.

For instance, Tsin and Kanade [1] propose a kernel correlation based point set registration approach where the cost function is proportional to the correlation of two kernel density estimates. Jian and Vemuri [2] instead resolve the problem representing each of the point by a mixture of Gaussians and treating the point set registration as a problem of aligning the two mixtures. Finally, Myronenko et al. [20] propose a probabilistic method for non-rigid registration which is treated as a Maximum Likelihood estimation problem with motion coherence constraint over the velocity field.

The reason for the popularity of probabilistic techniques stems from the fact that robot mapping is characterized by uncertainty and sensor noise, but the problem is that they have an intrinsic computational inefficiency, i.e. they are too slow for most realtime applications.

In this paper, we present a method that belongs to the firstly aforementioned class of approaches but with the main contributions that (i) it does not need an initial estimate of the relative pose, (ii) it works in very bad conditions (large amounts of noise and outliers), (iii) it has a very short convergence time and (iv) it has low error probability even with the 80% of outliers.

The thesis is organized as follows. In the Chapter 2 we discuss the issues involved in range image registration, including the definition of registration and our lidar model description. In the following chapters, we discuss the methods to achieve registration with respective results. We present the results obtained with the ICP registration and RANSAC-based registration (Chapter 3), then

we present our new proposed ICPSAC registration (Chapter 4) with some simulated and experimental results, finally we conclude by showing simulated results obtained by using some of the already existing approaches, the Coherent Point Drift [20] (CPD) and Kernel Correlation [1] (KC) algorithms, on the same data set utilized for testing ICPSAC (Chapter 5).

# Chapter 2

# LIDAR Registration Algorithm

## 2.1 Problem Definition

We address the question of building reconstruction and pose estimation combining LIDAR data only, obtained from several viewpoints, into a map using global registration.

LIDAR is an acronym for LIght Detection And Ranging. The lidar instrument is an optical remote sensing technology which transmits light out to a target and which measures properties of scattered light to find range and/or other information of a distant target. The transmitted light interacts with and is changed by the target. Some of this light is reflected/scattered back to the instrument where it is analysed. The change in the properties of the light enables some property of the target to be determined as well. Range finder lidars are the simplest lidars. They are used to measure the distance from the lidar instrument to a solid or hard target. The prevalent method to determine distance to an object or surface is to use laser pulses. Like the similar radar technology, which uses radio waves instead of light, the range to an object is determined by measuring the time delay between transmission of a pulse and detection of the reflected signal.

The reason why we want to deal with this problem is that computational efficiency is an important issue to take into account. In fact, the environment is sensed on-the-fly, i.e. while the robot is moving, and the robot sensors are subject to strict range limitations that make it necessary to navigate through its environment while building a map. A complicating aspect of the robot mapping problem arises from the high dimensionality of the entities that are being mapped. In such case, then, sensors which give a wide angle representation at a high frequency, even if the data is sparse, are favoured over a high resolution representation, which takes a longer acquisition time. Lasers sen-

sors provide range images* consisting of a set of point-measurements. Usually, such range images are acquired from several viepoints by moving the entire setup through an environment and measuring with a 2D laser orthogonally to the motion trajectory. For sensing the environment SICK LMS 291 laser range finders are used on our robotic platform. With such a configuration the question of determining the displacement of the setup arises. In the following we focus on localization and map building.

Most of the work in mobile robotics regarding these topics is based on two-dimensional data, where the robot is assumed to move on flat ground. Research in mobile robotics is concerned with finding solutions to the problem of localization and map building which enable the robot to autonomously localize itself and create a map. To create globally consistent maps it is necessary for the robot to be continously localized while registering new data with the map is building. This is known in the literature as *Simultaneous Localization and Mapping* (SLAM) problem. This is because mapping includes both, estimating the position of the robot relative to the map and generating a map using the sensory input and the estimates about the robot's pose. The ability to simultaneously localize a robot and accurately map its environment is considered by many to be a key prerequisite of truly autonomous robots but it is considered to be complex problem as well, because for localization a robot needs a consistent map and for acquiring the map the robot requires a good estimate of its location. This mutual dependency among the pose and the map estimates makes the SLAM problem hard.

## 2.2 LIDAR Model

### 2.2.1 Range of apllication and product features

SICK LMS 291 laser scanner is non-contact measurement system, that scan its surroundings two-dimensionally. As scanning system, the device require neither reflectors nor position marks. The system can be used in a bunch of applications:

- determining the volumes or contours of bulk materials;
- determining the volumes of objects;
- determining the positions of objects;
- collision prevention.

*In our context, by *image* we mean the surface information of the object or of the environment from specific point of view.

In our context, the LMS 291 is prevalently used for determining the position of the objects in the environment. In fact the scanner's measurement data are individually processed in real time with an external evaluation software for *determining the position*.

The advantages in using the LMS 291 are:

- a rapid scanning times, thus measurements objects can move at high speed;
- no special target-object reflective properties are necessary;
- no reflectors and no marking of the measurement object are necessary;
- backgrounds and surroundings do not have any influence on the measurements;
- measurement objects may be in any position;
- measurement data is available in real time and can be used for further processing via a serial interface;
- no illumination of the measurement area is required.

### 2.2.2 Operating principle

The LMS system operates by measuring the *time of flight* of laser light pulses: a pulsed laser beam is emitted and reflected if it meets an object surface within the range of the sensor. The reflection is registered by the scanner's receiver. The elapsed time between transmission and reception of the impulse is directly proportional to the distance between the scanner and the object. This measurement principle has the benefit to detect an object independently of object color and surface structure.

This range finder has a configurable angular resolution. A light impulse (spot) is emitted every 0.25°, 0.5° or 1°, depending on the variant, with a measuring range up to 80 meters and a maximum scanning angle of 180°.

The pulsed laser beam is deflected by an internal rotating mirror so that a fan-shaped scan is made of the surrounding area (laser radar). The contour of the target object is determined from the sequence of impulses received. Laser scanner measurement data is used for object measurement and determining position. These measurement data correspond to the surrounding contour scanned by the device and are given out in binary format via the RS-232 interface. Fundamentally, the distance value per individual impulse is evaluated. This means that a distance value is provided every 0.25°, 0.5° or 1°, depending

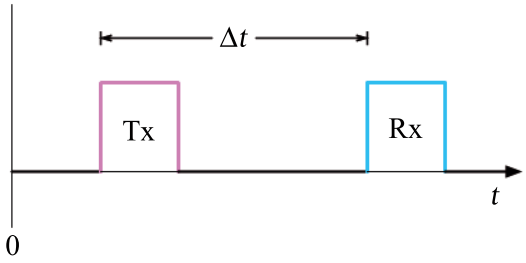



**Figure 2.1:** Time of flight between the transmission and reception of the impulse.

| | **Angular Resolution** | | |
|---|---|---|---|
| **Max. Scanning Angle** | **0.25°** | **0.5°** | **1°** |
| **180°** | 721 | 361 | 181 |
| **90°** | 361 | 181 | 91 |

**Table 2.1:** Maximum number of measured values for some different LIDAR's configurations.

on the angular resolution of the scanner. Note that the LMS turns towards the left (see Figure 2.2). For the simulation, LMS 291's noise standard deviation is used, whereas for the experimental tests the LMS 291 is configured in order to have an angular resolution of 0.5° and a scanning angle of 90°, so that 181 measured data points are obtained after each scan.

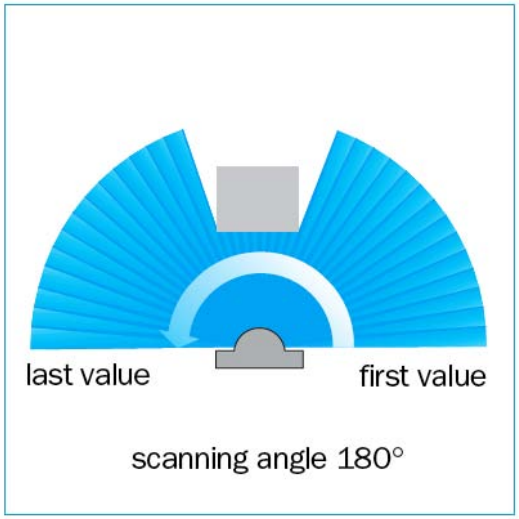



**Figure 2.2:** Direction of transmission for LMS 291 scanner.

| **Description** | **Value** |
|---|---|
| Range | max. 80 m |
| Angular resolution | 0.25°/0.5°/1° (selectable) |
| Response time | 53 ms/26 ms/13 ms |
| Measurement resolution[a] | 10 mm |
| Systematic error[b] | typ. ±35 mm (indoor mode) |
| Noise's standard deviation[c] ($\sigma_n$) | typ. 10 mm (indoor mode) |

[a]The ***resolution*** of a measuring device is the smallest possible distance different from zero between two consecutive individual measurement values. The resolution can be reduced by using averaged values.

[b]The ***systematic error*** is the sum of all the deviations over a defined excent of range and reflectivity, which cannot be reduced even using averaged values.

[c]The ***noise's standard deviation*** is calculated using a certain number of measurement values of a target with a certain reflectivity at a certain distance with a certain amount of illumination (at range $1, \ldots, 20$ m/$\geq$ 10% reflectivity/$\leq$ 5 kLux for the LMS 291).

**Table 2.2:** General Technical Data of the LMS 291.

## 2.3 Images Registration

Typically, a cloud of point samples from the surface of the objects is obtained from two or more points of view, in different reference frames. Let's consider two sets of points, the *model* set and the *scene* set which for convenience we will denote by $\mathcal{M}$ and $\mathcal{S}$, belonging to $\mathbb{R}^n$, where $n$ is usually equal to 3 or 2[†], with their elements being denoted by $\{\mathbf{m}_i\}_{i=1}^{N_m}$ and $\{\mathbf{s}_i\}_{i=1}^{N_s}$. The task of point set registration is to determine the parameters of a *rigid-body displacement*, denoted by

$$g \,:\, \mathbb{R}^n \to \mathbb{R}^n\,; \quad \mathbf{x} \mapsto g(\mathbf{x})\,, \tag{2.1}$$

which yields the best alignment of the two sets. In particular, the rotation matrix $R$ and the translation vector $T$ which, when applied to the scene points, best align model end scene.

The map $g$ preserves the norm and the cross product of any two vectors,

1. *norm*: $\|g_*(\vec{v})\| = \|\vec{v}\|\,, \quad \forall \vec{v} \in \mathbb{R}^n$, where

$$\vec{v} = \mathbf{x} - \mathbf{y}\,, \quad g_*(\vec{v}) := g(\mathbf{x}) - g(\mathbf{y})\,;$$

2. *cross product*: $g_*(\vec{v}) \times g_*(\vec{u}) = g_*(\vec{v} \times \vec{u})\,, \quad \forall \vec{u},\, \vec{v} \in \mathbb{R}^n$.

The set of all possible configurations of a rigid body can then be described by the space of rigid-body motions or special Euclidean transformations

$$SE(3) := \{g = (R, T) \mid R \in SO(3),\ T \in \mathbb{R}^3\}. \tag{2.2}$$

If we denote the actual transformation from the model to the scene configuration by $g_{sm} = (R_{sm}, T_{sm})$, i.e. indicating with $\mathbf{x}_m^{\mathbf{q}}$ the coordinates of a generic point $\mathbf{q}$ with respect to (w.r.t.) the model reference frame, its coordinates expressed w.r.t. the scene reference frame will be

$$\mathbf{x}_s^{\mathbf{q}} = g_{sm}(\mathbf{x}_m^{\mathbf{q}}) = R_{sm}\mathbf{x}_m^{\mathbf{q}} + T_{sm}\,. \tag{2.3}$$

Our goal is to find the robot's trajectory that, for the 2-D case, is at each step given by the orientation angle and the translation vector of the robot

$$\begin{cases} \theta^r = \theta_{sm} \\ T^r = T_{ms} \end{cases}. \tag{2.4}$$

[†]In our specific case, we will centre our attention to $n = 2$ case for the reason why lidars measure on a plane.

For solving this problem, we need to find, therefore, the transformation $g_{sm}$ or the inverse $g_{sm}^{-1} = g_{ms} = (R_{ms}, T_{ms})$, being the two transormation linked by the following relations

$$\begin{cases} R_{ms} = R_{sm}^{-1} = R_{sm}^{T} \\ T_{ms} = -R_{sm}^{T} T_{sm} \end{cases} . \tag{2.5}$$

For instance, let $\mathbf{a}_{ms} \in \mathbb{R}^p$ be the parameter vector of the actual transformation from the scene to the model configuration. Then, the problem is solved if we are able to find the transformation $g_{ms}$ such that

$$\mathcal{M} = g_{ms}(\mathbf{a}_{ms}; \mathcal{S}) . \tag{2.6}$$

In fact, in such a case, we can find the unknown parameters, i.e. the robot's trajecotory, that for the 2-D case are given by the following relations

$$\begin{cases} \theta^r = \theta_{sm} = -\theta_{ms} \\ T^r = T_{ms} \end{cases} , \tag{2.7}$$

where $\theta_{ms}$ can be easily obtained from the rotation matrix $R_{ms}$.

To deal with our kind of approach, in order to correctly align the two sets, we need to establish a consistent point-to-point correspondence between the two sets, also known as *correspondence problem* or *data association problem*. The correspondence problem is the problem of determining not only if sensor measurements taken at different points in time correspond to the same physical object in the world, but also to determine if a specific measure taken at an instant of time corresponds to any among the previous measures, and if yes, what. Therefore, we need to specify a function of the data that find the true correspondence between the scene and the model points. For the moment, let's assume we know how to solve the latter problem and let's define the function

$$\phi \, : \, \mathbb{R} \rightarrow \mathbb{R} \, ; \quad \phi(i) \mapsto j \, , \tag{2.8}$$

that select for each point of the scene the corresponding model point.

Because of the outliers presence, a large portion of the scene points may have no correspondence in the model set. Yet, the numbers of points in the two sets might be different, that is $N_m \neq N_s$. If we indicate with $\eta$ the percentage of the data points that can be paired, then $N_{or} = \eta N_s$ points belong to the *overlapping region*[‡]. To cope with this problems, we introduce the weights

[‡]Let's assume the overlapping region is characteristic enough to allow for unambigous matching.

vector $\mathbf{w}$ whose $i$-th component $w_i$ is zero for no matching, one otherwise. Then the function to be minimized is

$$\min_{\mathbf{a}_{ms}} e(\mathbf{a}_{ms}, \phi) = \min_{\mathbf{a}_{ms}} \sum_{i=1}^{N_s} w_i \left\| \mathbf{x}_m^{\mathbf{m}_{\phi(i)}} - g_{ms}(\mathbf{a}_{ms}; \mathbf{x}_s^{\mathbf{s}_i}) \right\|^2 , \tag{2.9}$$

For a 2-D Euclidean registration, $p = 3$. Indeed, the parameters of $g_{ms}$ are the rotation angle $\theta_{ms}$ and the two elements of the translation vector $T_{ms} = [t_{ms}^x \; t_{ms}^y]^T \in \mathbb{R}^2$. Then, in general, the coordinates $\mathbf{x}_s^{\mathbf{q}}$ of any point $\mathbf{q}$ expressed w.r.t the scene reference frame $< S >$ is transformed into the new coordinates $\mathbf{x}_m$ w.r.t. the model reference frame $< M >$ according to the following relation

$$\mathbf{x}_m^{\mathbf{q}} = g_{ms}(\mathbf{a}_{ms}; \mathbf{x}_s^{\mathbf{q}}) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \mathbf{x}_s^{\mathbf{q}} + \begin{bmatrix} t_{ms}^x \\ t_{ms}^y \end{bmatrix} , \quad \mathbf{x}_m^{\mathbf{q}} , \; \mathbf{x}_s^{\mathbf{q}} \in \mathbb{R}^2 . \tag{2.10}$$

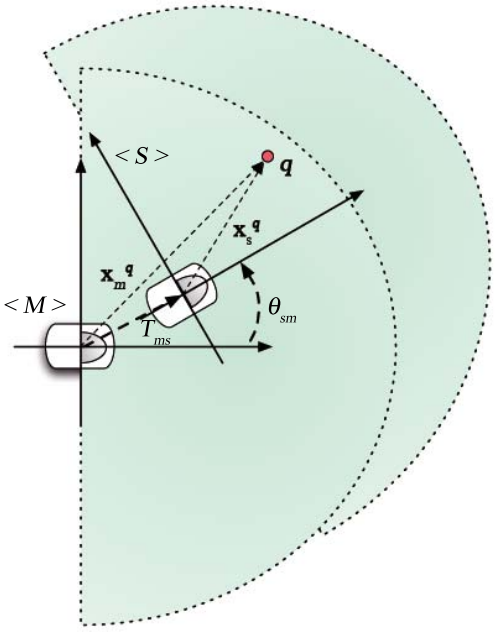



**Figure 2.3:** Coordinate frames of two consecutives scans for the 2-D case. $< M >$ is the reference frame of the first scan, $< S >$ the reference frame of the second one.

For a 3-D Euclidean registration, the number of parameters is $p = 6$ and they are the three rotation angles $\alpha_{ms}$, $\beta_{ms}$, $\gamma_{ms}$ along the three main axis

and the three elements of the translation vector $T_{ms} = [t^x_{ms}\ t^y_{ms}\ t^z_{ms}]^T \in \mathbb{R}^3$. Then the corresponding (2.10) is

$$\begin{aligned} \mathbf{x}^{\mathbf{q}}_m &= g_{ms}(\mathbf{a}_{ms}; \mathbf{x}^{\mathbf{q}}_s) \\ &= \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \mathbf{x}^{\mathbf{q}}_s + \begin{bmatrix} t^x_{ms} \\ t^y_{ms} \\ t^z_{ms} \end{bmatrix}, \quad (2.11) \end{aligned}$$

where $\mathbf{x}^{\mathbf{q}}_m,\ \mathbf{x}^{\mathbf{q}}_s \in \mathbb{R}^3$ and $cx$ and $sx$ stand for $\cos x$ and $\sin x$ respectively.

In general, the affine relations (2.10) and (2.11) can be rewritten in a linear form by using the *homogeneous representation*, i.e. if $g = (R, T),\ R, T \in \mathbb{R}^n$, then its homogeneous representation is

$$\bar{g} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+1)} \quad (2.12)$$

with which we can represent the transformation by a linear matrix multiplication. In such case, the (2.10) and (2.11) can be written as

$$\mathbf{X}^{\mathbf{q}}_m = \bar{g}_{ms}\mathbf{X}^{\mathbf{q}}_s\,, \quad (2.13)$$

where $\mathbf{X}^{\mathbf{q}}_m$ is the homogeneous representation of $\mathbf{x}^{\mathbf{q}}_m$, that is

$$\mathbf{X}^{\mathbf{q}}_m = \begin{bmatrix} \mathbf{x}^{\mathbf{q}}_m \\ 1 \end{bmatrix} \in \mathbb{R}^{n+1}\,. \quad (2.14)$$

We conclude this section punctualizing the fact that the relation (2.7) gives the robot's orientation and position at each step if the robot's reference frame coincide with the lidar's one. If this is not the case, for finding the real robot's configuration, we need to express the estimated transformation $g_{ms}$ w.r.t. the robot's reference frame. Since the lidar is fixed on the robot's chassis, it is sufficient to take the transformation $g_{rl}$, i.e. the transformation from the lidar's reference frame $< L >$ to the robot's reference frame $< R >$, into account as well. The transformation $g_{rl}$ is known and using the homogeneous representation it can be written as

$$\bar{g}_{rl} = \begin{bmatrix} R_{rl} & T_{rl} \\ 0 & 1 \end{bmatrix}, \quad (2.15)$$

therefore the robot configuration can be obtained, using the homogeneous representation, by simply premultiplying the matrix $\bar{g}_{ms}$ by $\bar{g}_{rl}$

$$\bar{g}^r_{ms} = \bar{g}_{rl}\bar{g}_{ms}\,. \quad (2.16)$$

## 2.4 Global Registration

To digitalize the environment, multiple scans have to be registered. If we denote with $< L^i >$ the reference frame of the $i$-th lidar scan, in order to initialize the global registration problem, we can think of the first lidar scan as the initial *model* reference frame and of the second scan as the initial *scene* reference frame, i.e. at the step $k = 0$ we have $< M >_0 = < L^1 >$ and $< S >_0 = < L^2 >$. Then, for a generic step $k$, the model frame $< M >_k$ and the scene frame $< S >_k$ will be given by

$$\begin{cases} < M >_k = < S >_{k-1} = < L^{k+1} > \\ < S >_k = < L^{k+2} > \end{cases} , \quad k = 1, 2, 3, \ldots \tag{2.17}$$

After registration, the scene has to be globally consistent. Various methods have been proposed for minimizing the trajectory error and then improving the overall result. A straightforward method is the *pairwise matching*, where the new scan is registered against the scan with the largest overlapping areas. Instead, Chen and Medioni introduced an *incremental matching* method. The new scan is registered against the union of the previously acquired and registered scans, i.e., when we merge a current view range image, instead of registering it with only a neighboring view, we can register it with the merged data from all previously processed views to find out the needed transformation. Both methods accumulate the registration errors such that the registration of many scans leads to inconsistent scenes and to problems with the robot localization. Other matching approaches with global error minimization have been proposed, e.g., Pulli [8] presents a registration method that minimizes the global error and avoids inconsistent scenes, distributing the global error while the registration of one scan is followed by registration of all neighboring scans.

# Chapter 3

# Preliminary Work

Before discussing about our new approach, we show in this chapter the limits of the ICP and RANSAC based approaches and what motivated us to look for improvements.

## 3.1 ICP Algorithm

### 3.1.1 Algorithm description

The *Iterative Closest Point* (ICP) algorithm is one of the most common approaches to image registration problem because of its semplicity and performance. ICP starts with two meshes and an initial guess for their relative rigid body transformation, and iteratively refines the transformation repeatedly generating pairs of corresponding points and minimizing an error metric.

Beginning with an initial estimate of the registration parameters, $\mathbf{a}_{ms}^0$, the algorithm forms a sequence of estimates $\mathbf{a}_{ms}^k$ which progressively reduce the error $e(\mathbf{a}_{ms})$ defined in the (2.9). Each iteration of the algorithm comprises the following three steps:

1. Compute correspondences $\phi$, pairing each point of $\mathcal{S}$ to the *closest* point in $\mathcal{M}$:

$$\phi(i) = arg \min_{j \in \{1 \dots N_m\}} \left\| \mathbf{x}_m^{\mathbf{m}_j} - g_{ms}(\mathbf{a}_{ms}^k; \mathbf{x}_s^{\mathbf{s}_i}) \right\|^2, \quad i = 1 \dots N_s, \tag{3.1}$$

   so that $\mathbf{m}_{\phi(i)}$ is the closest model point to the point $\mathbf{s}_i$ transformed by the current estimate $\mathbf{a}_{ms}^k$.

2. Update the transformation, $\mathbf{a}_{ms}^k$, computing the motion that minimizes

the mean square error between the paired points:

$$\mathbf{a}_{ms}^{k+1} = arg\min_{\mathbf{a}_{ms}} \sum_{i=1}^{N_s} \left\| \mathbf{x}_m^{\mathbf{m}_{\phi(i)}} - g_{ms}(\mathbf{a}_{ms}^k; \mathbf{x}_s^{\mathbf{s}_i}) \right\|^2, \tag{3.2}$$

This step can be performed in closed form procedures such as Singular Value Decomposition (SVD) or the quaternion-based algorithm*.

3. Apply the motion to $\mathcal{S}$ and update the *mean squared error* (MSE).

The three steps are iterated untill the set of correspondences does not change in the first step, in fact the value of $\mathbf{a}_{ms}^{k+1}$ would be set equal to $\mathbf{a}_{ms}^k$ in the second step, so no further change is possible. Convergence to a local minimum is guaranteed.

Anyway, this algorithm it has its own limitations. The local convergence problem requires sufficient overlap between the data-sets and a close inizialization in order to be able to find the true transformation. Also, a naive implementation of ICP is prone to failure due to outliers. We shows some experimental results in the next section.

### 3.1.2 Performance evaluation

For testing the algorithm, we perform some tests on synthetic data, with different configuration and working conditions. To form $\mathcal{S}$, the original model $\mathcal{M}$ is rotated by a known angle. Then, noise and uniformly distributed outliers are added to both the data sets. In the following figures, on the left is the model $\mathcal{M}$ and superimposed unregistered scene $\mathcal{S}$; on the right is the registered data after running the algorithm.

First, to test the ICP algorithm, we perform a set of exact rigid registration experiments without noise and outliers. The algorithm behaves well in ideal conditions with small rotation angle (Figure 3.1), but with larger values of $\theta$, e.g. $\theta = -\pi/4$ as in Figure 3.2, the result is not satisfactory at all.

Introducing a medium level of noise, it still continues to work, but it presents a larger sensitivity to the outliers (Figure 3.4); with 30% of outliers the resulting estimation is totally wrong.

Finally, testing ICP with data sets affected by both noise and outliers, the result is like we expected. ICP is not able to find the true transformation, but it converges in a local minimum.

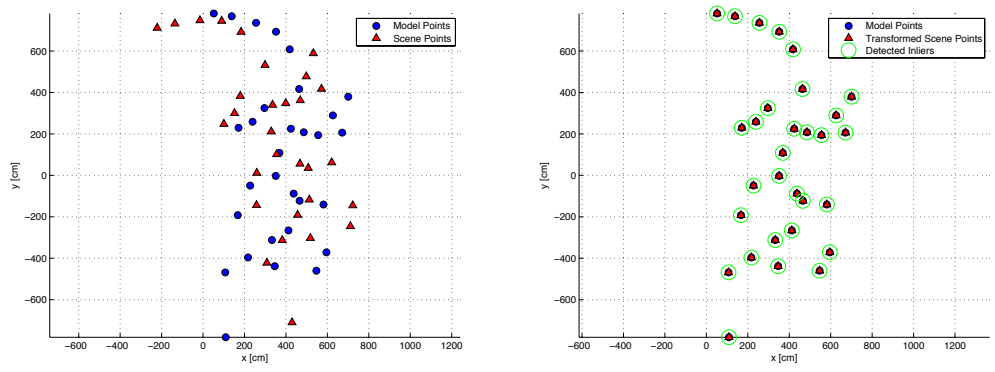
*See the Appendix A for more details.

### 3.1.3 Final considerations

Concluding this Section, we can state ICP works pretty fine as long as no outliers are present and for very small rotation angle. The Figure 3.7 shows the mean and the standard deviation of the errors on the angle $\theta$ and the translation vector $T$. The error is small only for small $\theta$ and no outliers.

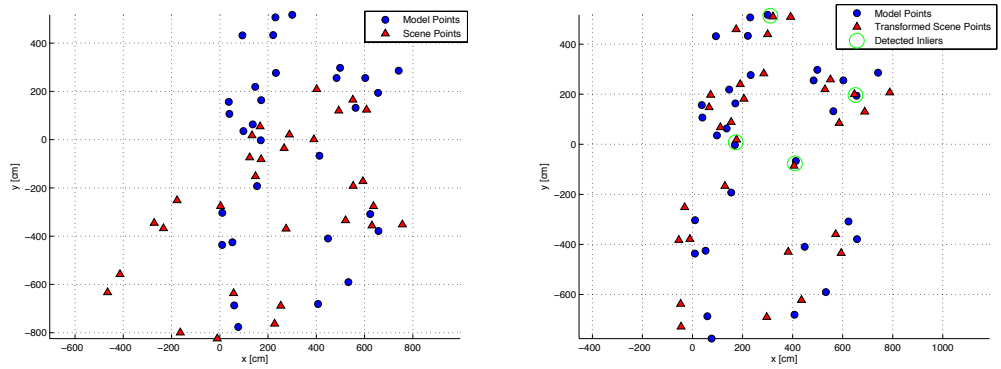
Thus, even if ICP is very simple and intuitive algorithm, it presents a large sensitivity to outliers and above all to medium values of $\theta$, local minima and exact point matching.

For this reason, since the introduction of ICP, there have been many improvements of it. Also, the computational expense of the ICP algorithm depends mainly on the number of points. In a brute force implementation the point pairing is in $O(n^2)$. Data reduction reduces the time required for matching. Many approaches have been introduced for subsampling the data. One of these was proposed by Thrun et al. [9]. They proposed a fast filtering method to reduce and smooth the data. It is a combination of a median and a reduction filter. The median filter removes the outliers by replacing a data point with the median value of the $n$ surrounding points. The data reduction instead joins into one point multiple data points located close together. Other improvements affect the nearest neighbor search. In fact at each iteration, corresponding points between the two images are determined by a nearest neighbor method. Determining these correspondences accounts for the majorty of the runtime expense of ICP. Besl and McKay [3] suggested the use of the *k-d tree* method that uses a backtracking search to identify nearest neighbors whose expected computation time to perform each search is proportional to $\log N$, where $N$ is the number of points. They also indicated that other methods may be suitable. There have been some explorations into the use of alternatives the most interesting one being the method proposed by Greenspan and Yurick [7] called *approximated k-d tree* that improves runtime efficiency with the tradoff of reducing the accuracy of the determined correspondences.



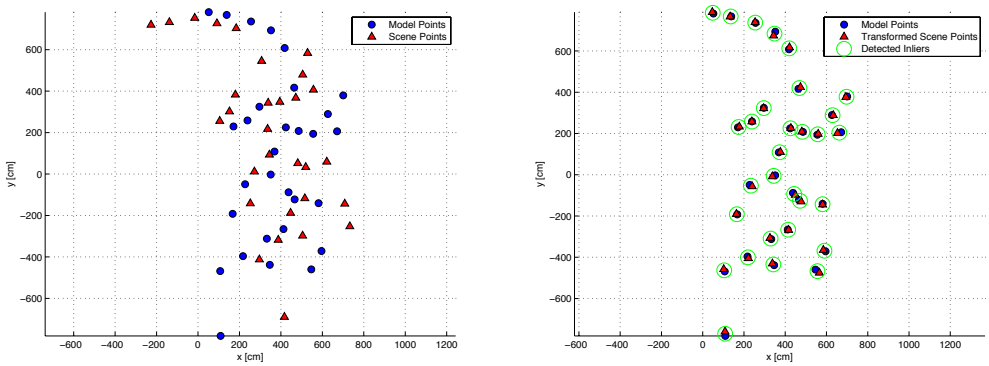

(a) (b)

**Figure 3.1:** ICP: (a) Initial configuration: $N_m = N_s = 30$, $\theta = \pi/8$ rad, $T = [30 \ -30]^T$ cm, no presence of noise and no outliers. (b) Registration result.



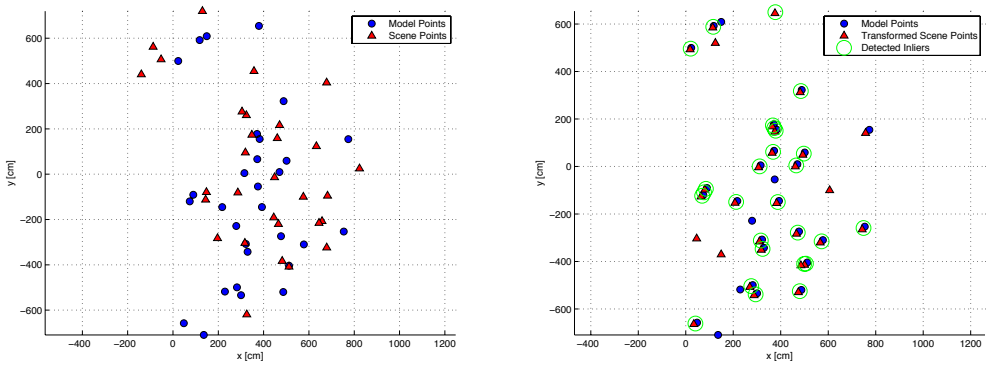

(a) (b)

**Figure 3.2:** ICP: (a) Initial configuration: $N_m = N_s = 30$, $\theta = -\pi/4$ rad, $T = [30 \ -30]^T$ cm, no presence of noise and no outliers. (b) Registration result.



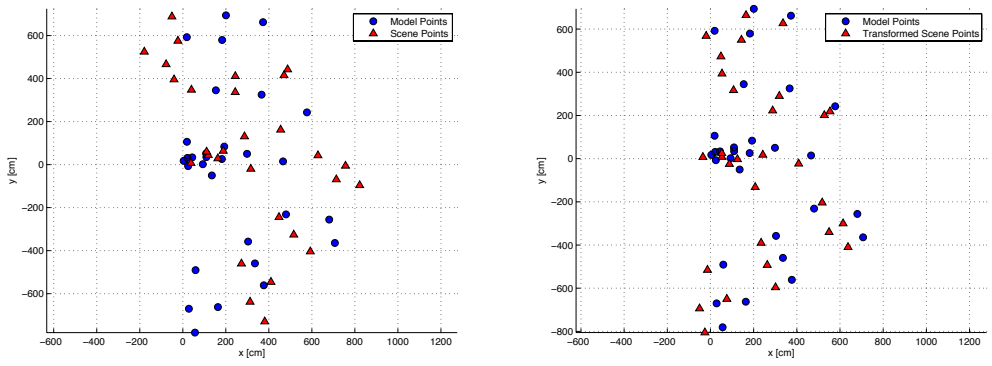

(a) (b)

**Figure 3.3:** ICP: (a) Initial configuration: $N_m = N_s = 30$, $\theta = \pi/8$ rad, $T = [30 \ -30]^T$ cm, presence of noise and no outliers. (b) Registration result.



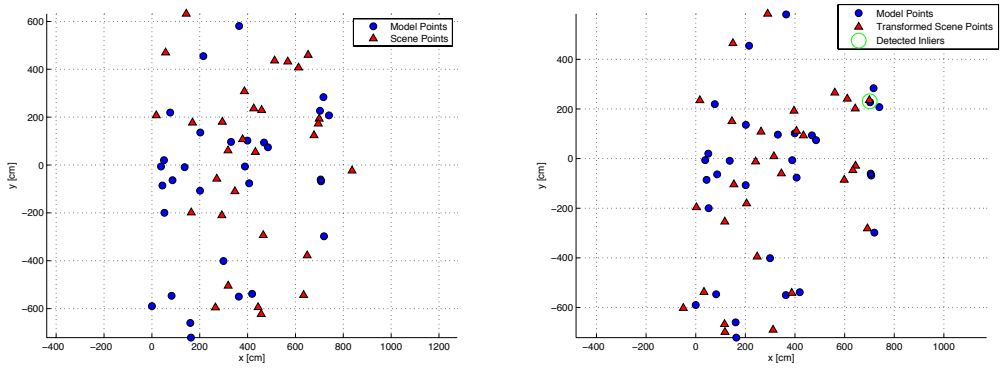

(a) (b)

**Figure 3.4:** ICP: (a) Initial configuration: $N_m = N_s = 30$, $\theta = \pi/8$ rad, $T = [30 \ -30]^T$ cm, no presence of noise and 20% of outliers. (b) Registration result.




(a) (b)

**Figure 3.5:** ICP: (a) Initial configuration: $N_m = N_s = 30$, $\theta = \pi/8$ rad, $T = [30 \ -30]^T$ cm, no presence of noise and 30% of outliers. (b) Registration result.




(a) (b)

**Figure 3.6:** ICP: (a) Initial configuration: $N_m = N_s = 30$, $\theta = \pi/8$ rad, $T = [30 \ -30]^T$ cm, presence of noise and 30% of outliers. (b) Registration result.

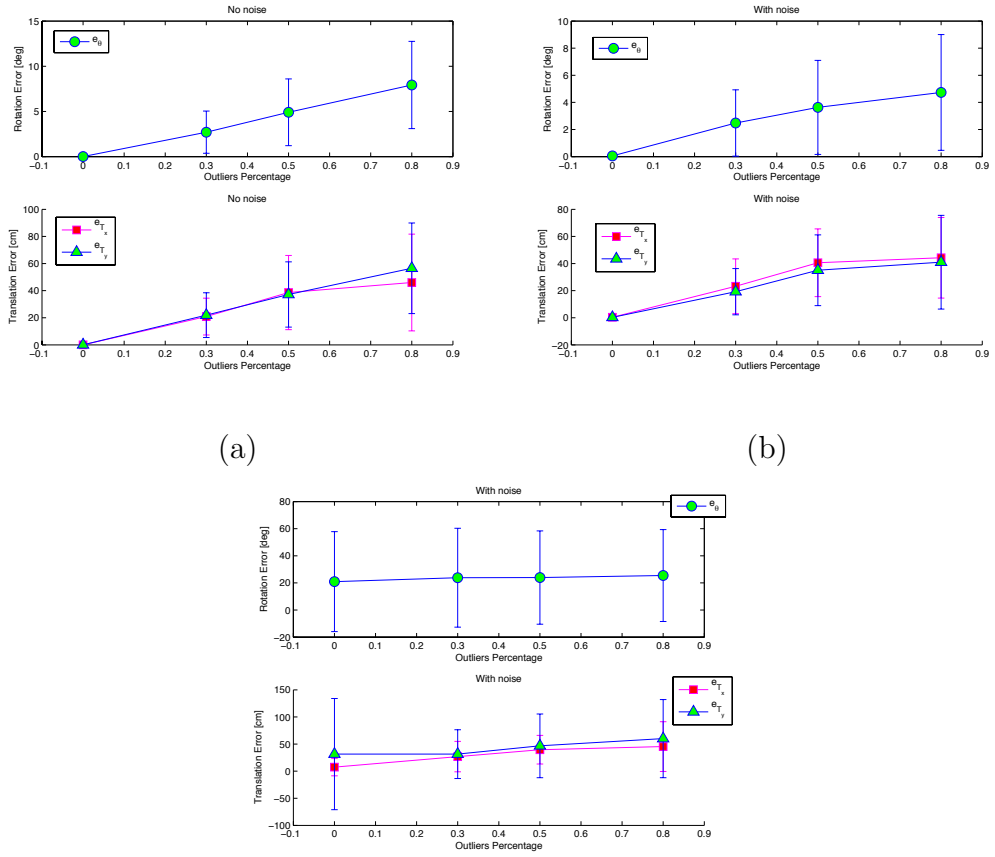



**Figure 3.7:** ICP: (a) Mean and standard deviation of the error on the estimated rotation angle $\theta$ and on the estimated translation vector $T$ without noise and small angle $\theta$ (up to $\pi/8$ rad). (b) As (a) but with noise presence. (c) Mean and standard deviation of the error on the estimated rotation angle $\theta$ and on the estimated translation vector $T$ with noise and different values of $\theta$ (up to $\pi/4$ rad).

## 3.2 RANSAC Based Registration

### 3.2.1 Algorithm description

The *Random Sample Consensus* (RANSAC) algorithm introduced by Fishler and Bolles [6] is a widely used robust estimator that has become a standard solutions in many estimation problems.

The RANSAC algorithm proceeds as follows. Repeatedly, subsets of the input data are randomly selected and model parameters fitting the sample are computed. In a second step, the quality of the parameters is evaluated on the input data. Different cost functions (see Section 4.3) have been proposed, the standard being the number of inliers, i.e. the number of data points consistent with the model. The process is terminated when the probability of finding a better model becomes lower than a probability $\xi$. The speed depends on two factors. The percentage of outliers determines the number of random samples needed to guarantee the $1 - \xi$ confidence in the solution. The time needed to access the quality of a hypothesized model parameters is proportional to the number of the input data points. Almost all models whose quality is verified are incorrect with arbitrary parameters originating from contaminated samples, therefore such models are consistent with only a small number of the data points.

The idea is to use this concepts for building a more stable to outliers algorithm that we call RANSAC-based algorithm. Each iteration of this algorithm comprises the following three steps:

1. A random sample of the minimum number of points among the scene set $\mathcal{S}$ is selected. We call this minimal set $\mathcal{S}_h$.

2. The ICP algorithm, as described in the Section 3.1, is applied with inputs $\mathcal{S}_h$ and $\mathcal{M}$.

3. The number of inliers is counted.

The three steps are iterated for a fixed number of trials and then the best result (e.g., the registration with the bigger number of inliers[†]) is given as output.

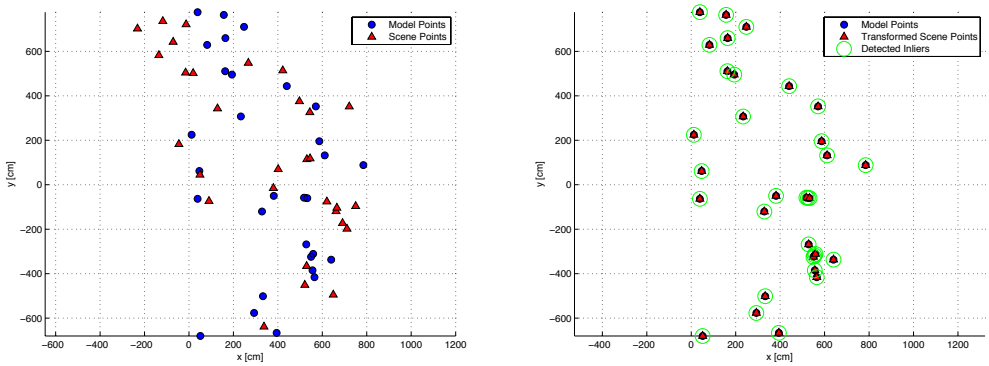
### 3.2.2 Performance evaluation

First, to test the RANSAC-based algorithm, we perform a set of exact rigid registration experiments without noise and outliers. We generate the reference model and the rotated model the same way as the previous section. We add

[†]For other possible loss function, see Section 4.3.

outliers points randomly drawn from a uniform distribution. On the left is the model $\mathcal{M}$ and superimposed unregistered scene $\mathcal{S}$; on the right is the registered data after running the algorithm.

The algorithm behaves well in ideal conditions with small rotation angle (Figure 3.8), but with larger values of $\theta$, e.g. $\theta = -\pi/4$, as shown in Figure 3.13.(c), the result is not as satisfactory as ICP's one. Also, it works well as long as the outliers percentage is low. In fact, with a percentage of 50%, it is no longer able to find the correct transformation (Figure 3.12).



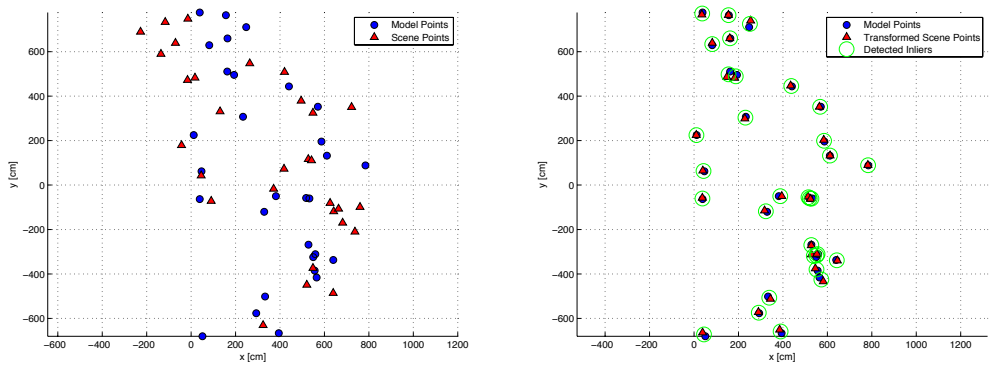

(a) (b)

**Figure 3.8:** RANSAC: (a) Initial configuration: $N_m = N_s = 30$, $\theta = \pi/8$ rad, $T = [30\ -30]^T$ cm, no presence of noise and no outliers. (b) Registration result.

### 3.2.3 Final considerations

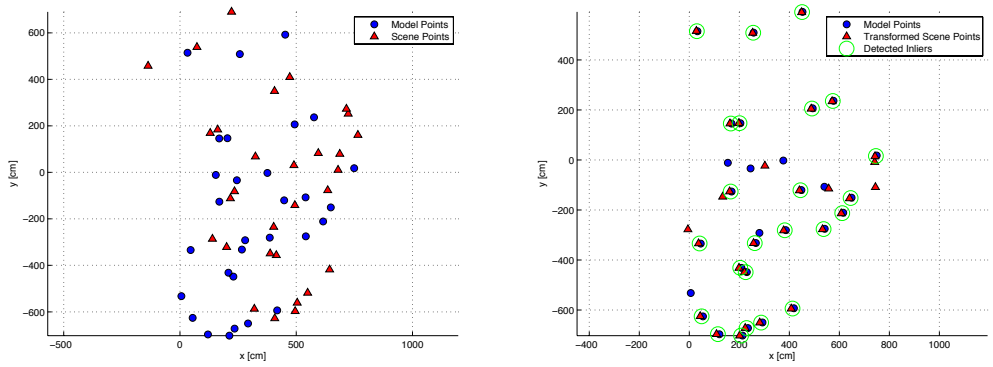
The algorithm seems to work well in presence of low outliers percentages (Figure 3.10-3.11) but by performing a test with rotation angles up to $\pi/4$ rad, the result is not satisfactory. In such case, the average error on the estimated rotation angle is over 20 degrees.

In the next chapter we explain how to improve this approach for obtaining better results with a more robust at outliers approach.



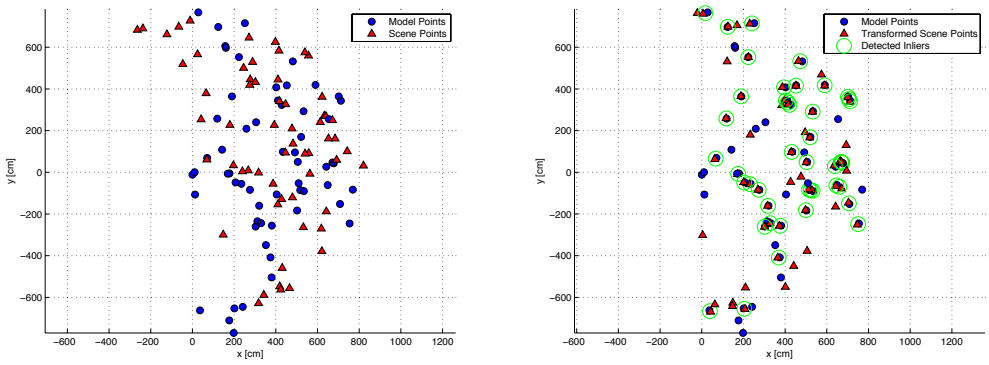

(a) (b)

**Figure 3.9:** RANSAC: (a) Initial configuration: $N_m = N_s = 30$, $\theta = \pi/8$ rad, $T = [30 \ -30]^T$ cm, presence of noise and no outliers. (b) Registration result.



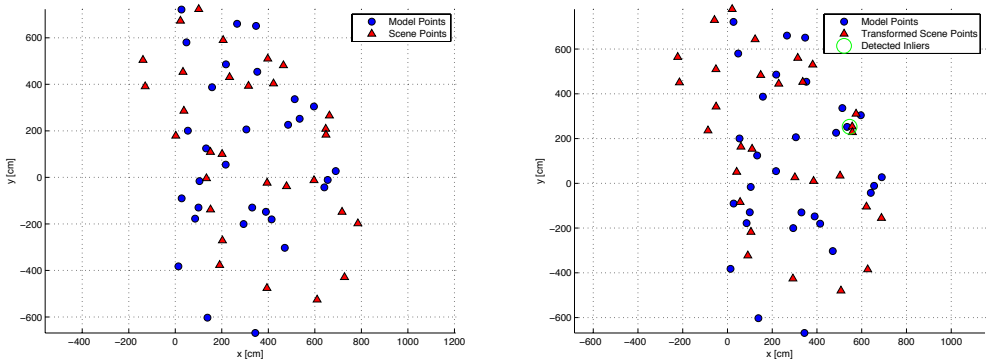

(a) (b)

**Figure 3.10:** RANSAC: (a) Initial configuration: $N_m = N_s = 30$, $\theta = \pi/8$ rad, $T = [30 \ -30]^T$ cm, no presence of noise and 20% of outliers. (b) Registration result.




(a) (b)

**Figure 3.11:** RANSAC: (a) Initial configuration: $N_m = N_s = 90$, $\theta = \pi/8$ rad, $T = [30 \ -30]^T$ cm, no presence of noise and 40% of outliers. (b) Registration result.




(a) (b)

**Figure 3.12:** RANSAC: (a) Initial configuration: $N_m = N_s = 30$, $\theta = \pi/8$ rad, $T = [30 \ -30]^T$ cm, presence of noise and 50% of outliers. (b) Registration result.

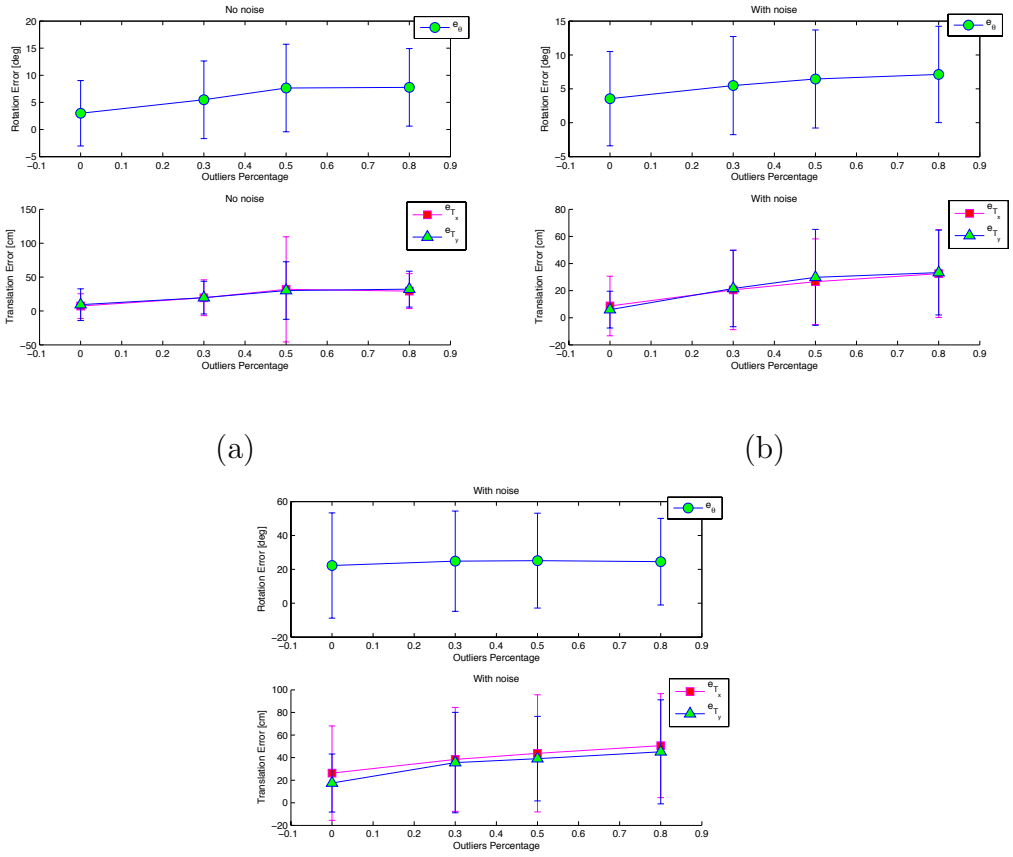



**Figure 3.13:** RANSAC: (a) Mean and standard deviation of the error on the estimated rotation angle $\theta$ and on the estimated translation vector $T$ without noise and small angle $\theta$ (up to $\pi/8$ rad). (b) As (a) but with noise presence. (c) Mean and standard deviation of the error on the estimated rotation angle $\theta$ and on the estimated translation vector $T$ with noise and different values of $\theta$ (up to $\pi/4$ rad).

# Chapter 4

# ICPSAC Registration

In this chapter, we introduce a new and largely improved algorithm for Point Set Registration by using a RANSAC-based ICP approach, called ICPSAC. The objective of ICPSAC is to find inliers in the shortest possible time and to guarantee, with a certain probability, that all the inliers and then the correct solution at the registration problem are found.

## 4.1 ICPSAC Algorithm Description

The basic idea of ICPSAC is to merge the ICP and the RANSAC well known concepts for constructing an algorithm much more robust to noise and outliers presence than any other current counterpart. The algorithm is more formally stated* in Table 4.1.

Our algorithm is very simple and it works in the following way. First of all, the mutual distances among all the points in the model and in the scene are computed. Then the algorithm described in Table 4.1 is applied as far as a sufficient number of inliers is founded. The algorithm could be run for a fixed number of trials and then one can take as output the best transformation among all the computed ones. In this way, one can avoid to be worried about the parameter $N_{min}$, but this approach might waste computational time. We assume in fact the noise is not large enough and that it's sufficient a small amount of inliers to detect the true transformation. In such a case, the computational time is privileged. We punctualize the fact that the reasearch of the nearest couple $(\mathbf{m}_1, \mathbf{m}_2)$ in the step 2 of the algorithm is of crucial importance for its good behaviour. With reference to the Figure 4.1, the nearest couple is found in this way. First of all, the nearest point $\mathbf{m}_1$ to the point $\mathbf{s}_1$, according

*For simplicity, the algorithm is stated for the 2-D case but it can be easily extended for the 3-D case.

1. **Selection of the *scene control points***

   (a) Select randomly the minimum number of points (2 points for a 2D transformation, see Section A.2), $\mathbf{s}_1$ and $\mathbf{s}_2$, among the set $\mathcal{S}$ with the constraint

   $$d_{\mathcal{S}} = d(\mathbf{s}_1, \mathbf{s}_2) \geq d_{min}$$

   where $d_{min}$ is a user controllable threshold set trading between noise effect reduction and correct registration probability enhancement (see Section 4.2.3).

2. **Selection of the *model control points***

   (a) Select, among all the points in $\mathcal{M}$, the *nearest* (see Section 4.1) couple $(\mathbf{m}_1, \mathbf{m}_2)$ to the previously selected couple $(\mathbf{s}_1, \mathbf{s}_2)$ with the constraint

   $$d_{\mathcal{M}} = d(\mathbf{m}_1, \mathbf{m}_2) \simeq d_{\mathcal{S}}\,.$$

   - If such a couple does not exist, go back to the step 1.

3. **Model Parameter Estimation**

   (a) Find the transformation parameters $\mathbf{a}$ using the selected couples.

   (b) Apply the estimated transformation to the set $\mathcal{S}$ for finding

   $$\mathcal{S}' = g(\mathbf{a}; \mathcal{S})\,.$$

4. **Model Verification**

   (a) Count the number $N_{in}$ of points in $\mathcal{S}'$ consistent with the model $\mathcal{M}$.

   - If $N_{in} \geq N_{min}$, where $N_{min}$ depends on the outliers percentage, then $\mathbf{a}$ is a good estimate; solve the least mean square problem only on the found inliers,

   $$\mathbf{a}^* = arg\min_{\mathbf{a}} \sum_{i=1}^{N_{in}} \left\| \mathbf{x}_m^{\mathbf{m}_i} - g(\mathbf{a}; \mathbf{x}_s^{\mathbf{s}_i}) \right\|^2,$$

   and return $\mathbf{a}^*$.

   - If $N_{in} < N_{min}$, go back to the step 1.

**Table 4.1:** A summary of the stages of the ICPSAC algorithm for a 2D transformation.

1. **Selection of the *scene control points***

   (a) Select randomly the minimum number of points (2 points for a 2D transformation, see Section A.2), $\bar{\mathbf{s}}_1$ and $\bar{\mathbf{s}}_2$, among the set $\mathcal{S}$ with the constraint

   $$d_{\mathcal{S}} = d(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2) \geq d_{min}$$

   where $d_{min}$ is a user controllable threshold set trading between noise effect reduction and correct registration probability enhancement (see Section 4.2.3).

2. **Selection of the *model control points***

   (a) Select as feasible all those couples of points whose mutual distance $d_{\mathcal{M}}$ is approximately equal to $d_{\mathcal{S}}$,

   $$\mathcal{C} = \{(\mathbf{m}_i, \mathbf{m}_j),\ i, j = 1, \ldots, N_m,\ i \neq j \mid d_{\mathcal{M}} = d(\mathbf{m}_i, \mathbf{m}_j) \simeq d_{\mathcal{S}}\},$$

   and

   $$\bar{\mathcal{C}} = \{\mathbf{m}_i \mid (\mathbf{m}_i, \mathbf{m}_j) \in \mathcal{C}\},$$

   being $N_c = \#\bar{\mathcal{C}}$.

   - If the set $\mathcal{C}$ is empty, go back to the step 1.(a).

   (b) Sort in ascending order all the points $\mathbf{c}_i$ belonging to the set $\mathcal{C}$ according to their distance from the point $\bar{\mathbf{s}}_1$,

   $$\mathcal{C}_{\bar{\mathbf{s}}_1} = \{\mathbf{c}_i,\ i = 1, \ldots, N_c \mid d(\mathbf{c}_i, \bar{\mathbf{s}}_1) < d(\mathbf{c}_{i+1}, \bar{\mathbf{s}}_1)\}.$$

   (c) For $i = 1$ to $N_c$, select $\bar{\mathbf{m}}_1 = \mathbf{c}_i$ as the corresponding point to $\bar{\mathbf{s}}_1$.

   - Select as the corresponding point to $\bar{\mathbf{s}}_2$ the point $\bar{\mathbf{m}}_2$ such that the couple $(\bar{\mathbf{m}}_1, \bar{\mathbf{m}}_2) \in \mathcal{C}$ and $d(\bar{\mathbf{m}}_2, \bar{\mathbf{s}}_2) < d(\mathbf{c}_j, \bar{\mathbf{s}}_2),\ \forall j = 1, \ldots, N_c, j \neq i$ and $(\bar{\mathbf{m}}_1, \mathbf{c}_j) \in \mathcal{C}$.
     - If a couple is found, go to the step 3.
     - If no couple is found, go back to the step 2.(c) where $i \leftarrow i + 1$.

3. **Model Parameter Estimation**

   (a) Find the transformation parameters $\mathbf{a}$ using the two selected couple

   (b) Apply the estimated transformation to the set $\mathcal{S}$ for finding

   $$\mathcal{S}' = g(\mathbf{a}; \mathcal{S})$$

4. **Model Verification**

   (a) Count the number $N_{in}$ of points in $\mathcal{S}'$ consistent with the model $\mathcal{M}$.

   - If $N_{in} \geq N^*$, then $N^* = N_{in}$ and $\mathbf{a}^* = \mathbf{a}$ where $\mathbf{a}^*$ is the best found solution and $N^*$ the greatest number of found inliers so far.
     - If $N^* \geq N_{min}$ where $N_{min}$ is a threshold that depends on the outliers percentage, then $\mathbf{a}^*$ is a good estimation; solve the least mean square problem only on the found inliers,

     $$\mathbf{a}^* = arg \min_{\mathbf{a}} \sum_{i=1}^{N_{in}} \left\| \mathbf{x}_m^{\mathbf{m}_i} - g(\mathbf{a}; \mathbf{x}_s^{\mathbf{s}_i}) \right\|^2,$$

     and return $\mathbf{a}^*$.
   - If $N^* < N_{min}$, then go back to the step 2.(c) where $i \leftarrow i + 1$.

**Table 4.2:** A summary of the stages of the modified ICPSAC algorithm for a 2D transformation.

to the euclidean distance, is selected. Then, among all the points of $\mathcal{M}$ that have distance from $\mathbf{m}_1$ approximately equal to the distance $d_{\mathcal{S}}$ between $\mathbf{s}_1$ and $\mathbf{s}_2$, i.e. the points in the green circular region whose width is set to $d_{th}$, the nearest to $\mathbf{s}_2$ is chosen to be the correspondent counterpart. It's obvious that, with reference to the Figure 4.1, $|d_{\mathcal{S}} - d_{\mathcal{M}}| \leq d_{th}/2$ and that $d_{th}$ depends from the strenght of the noise and therefore it should be appropriately set. In fact, if $d_{th}$ is too small, maybe no point is found. On the contrary, if $d_{th}$ is too large, it might happen that inconsistent points are found.

Finally, a more stable modified version of ICPSAC is stated in Table 4.2.

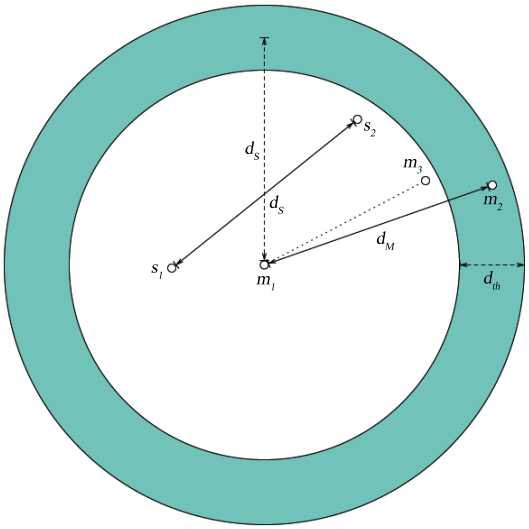



**Figure 4.1:** Selection of the control points in the 2D case: the $(\mathbf{m}_1, \mathbf{m}_2)$ couple is preferred on $(\mathbf{m}_1, \mathbf{m}_3)$.

## 4.2 Performance Evaluation

In this section we primarly discuss about the algorithm's performance and then we present some results on the application of our method to data sets.

### 4.2.1 Analysis of complexity

We briefly analyze here the computational complexity of the ICPSAC algorithm. Let's assume to have already computed the mutual distances among each point of the model and scene. Then the steps 1.(a) takes only $O(1)$. The step 1.(b) instead takes $O(N_m)$ because all the points in the model set $\mathcal{M}$ must

to be scored for ensuring to find the nearest points to the selected points in the scene set $\mathcal{S}$. However the complexity can be decreased to $O(\log N_m)$ using a *kd-tree*. Once the points are selected, finding the transformation between the two couple of points takes only $O(1)$, and then applying that to all the scene points takes $O(N_s)$. Also the count of the inliers takes $O(N_s)$. We can then state that the total computational time is $O(N)$, where $N = \max(N_m, N_s)$. This means that in the worst case, the computational time is linear in the number of the points in the scene or in the model.

Now, one can be asked how many iterations should the ICPSAC algorithm be run for. We discuss about this in the next Subsection.

### 4.2.2 Rate of convergence

Let's find now the number of trials $k$ required to select a subset of $n$ good data points in the scene and the right correspondent points in the model. Let $w$ be the probability to select these $n$ good data points. In the 2D case $n = 4$ and we have

$$\begin{aligned} w &= Pr(\mathbf{s}_1 \text{ is inlier}) \times Pr(\mathbf{m}_1 \text{ is inlier} \mid \mathbf{m}_1 \text{ is the closest point to } \mathbf{s}_1) \times \\ &\times Pr(\mathbf{s}_2 \text{ is inlier} \mid \mathbf{s}_1 \text{ is inlier}, d(\mathbf{s}_1, \mathbf{s}_2) = d_{\mathcal{S}}) \times \\ &\times Pr(\mathbf{m}_2 \text{ is inlier} \mid \mathbf{m}_2 \text{ is the closest point to } \mathbf{s}_2, d(\mathbf{m}_1, \mathbf{m}_2) = d_{\mathcal{M}} \simeq d_{\mathcal{S}}) \end{aligned} \tag{4.1}$$

It's easy to realize that it would be hard to analitically find the probability $w$. For this reason, we prefer to find it via experimental results (see Table 4.3). The probability distribution of the number $k$ of *Bernoulli trials*[†] needed to get one success has, in probability theory and statistics, a geometric distribution. If the probability of success on each trial is $p$, then the expected value and the variance of the geometrically distributed random variable $k$ are

$$E(k) = \frac{1}{p} \quad , \quad var(k) = \frac{1-p}{p^2} \quad . \tag{4.2}$$

Therefore, in our case, the number of attempts to find a consensus set is on average $1/w$ with a standard deviation of $\sqrt{1-w}/w$. From another point of view, if we want to know the expected selection number $k$ that ensures with probability $q$ that at least one of the random selections is a good choice for finding the true transformation, we should expect to wait for $k$ iterations, where $k$ is given by

$$(1 - (1-w)^k) = q \qquad \Longrightarrow \qquad k = \frac{\log(1-q)}{\log(1-w)} \; . \tag{4.3}$$

[†]A Bernoulli trial is an experiment whose outcome is random and can be either of two possible outcomes, "success" and "failure".

| $\theta$ (rad) | $\%_{out} = 0$ | 0.3 | 0.5 | 0.8 |
|---|---|---|---|---|
| 0 | 1.000 | 0.474 | 0.242 | 0.039 |
| $\pi/8$ | 0.181 | 0.093 | 0.050 | 0.013 |
| $\pi/5$ | 0.109 | 0.063 | 0.041 | 0.004 |
| $\pi/4$ | 0.086 | 0.058 | 0.023 | 0.002 |

(a)

| $\theta$ (rad) | $\%_{out} = 0$ | 0.3 | 0.5 | 0.8 |
|---|---|---|---|---|
| 0 | 0.978 | 0.479 | 0.264 | 0.026 |
| $\pi/8$ | 0.042 | 0.009 | 0.010 | 0.002 |
| $\pi/5$ | 0.026 | 0.010 | 0.009 | 0.002 |
| $\pi/4$ | 0.011 | 0.004 | 0.007 | 0.001 |

(b)

**Table 4.3:** (a) Typical values of $w$ for different values of $\theta$ and amount of outliers in the noiseless case. (b) Typical values of $w$ for different values of $\theta$ and amount of outliers in the noise case.

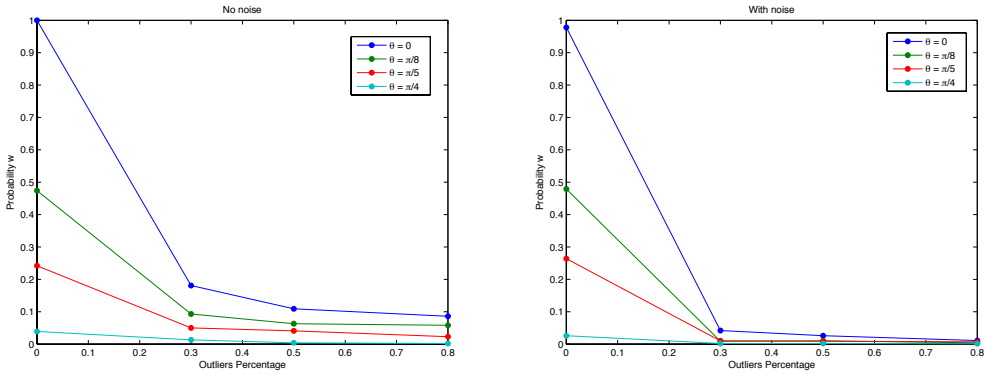
In Table 4.3 some experimentally computed values of $w$ for corresponding values of $\theta$ and outliers percentage are shown. For example, if we are in a usual configuration, i.e. presence of noise with 30% of outliers and a rotation angle $\theta = \pi/8$ we have $w \simeq 0.042$ and to obtain a 90% assurance of making at least one error-free selection, we might expect to find it in $k = \log(1 - 0.9)/\log(1 - 0.042) \simeq 53$ selections.

### 4.2.3 Sensitivity to noise

We use a method to test how sensitive the registration methods are in the presence of noise perturbation. We generate slightly rotated versions of the same point set and add zero mean random noise to both the reference model and the rotated model

#### Choice of the distance $d_{min}$

To acquire a map, robots must possess sensors that allow them to perceive the surrounding world. Sensors commonly used are cameras, range finders using sonars, laser and infrared technology, radar, GPS, etc. However, all




(a) (b)

**Figure 4.2:** (a) Typical values of $w$ for different values of $\theta$ and amount of outliers in the noiseless case. (b) Typical values of $w$ for different values of $\theta$ and amount of outliers in the noise case.

these sensors are subject to errors, often referred to as measurement noise.

We can think of the laser's measurements generated according to the following model

$$\mathbf{y}_t = f(\mathbf{a}_t, \mathbf{n}_t), \qquad t = 1, 2, 3, \ldots \tag{4.4}$$

that is, the measurements obtained at the time $t$ are a function of the parameter vector $\mathbf{a}_t$ and of the noise $\mathbf{n}_t$.

Let's assume that the uncertainty on the measurements can be expressed, according to the Gauss Error Theory Model, as an additive noise. Therefore, the result of the physical measurement comprises two parts: an estimate of the true value of the measurements and the uncertainty of its estimate. Under this assumption, the (4.4) can be written as

$$\mathbf{y}_t = f(\mathbf{a}_t) + \mathbf{n}_t, \qquad t = 1, 2, 3, \ldots \tag{4.5}$$

where $f(\mathbf{a}_t)$ is the ideal data from the instrument and $\mathbf{n}_t$ is the measurement error. Let's assume the Central Limit Theorem is valid in this contest and that the instrument is not affected by systematic errors (i.e., that can be removed by suitable calibration). Then we can think that $\mathbf{n}_t$ is normally distributed with zero mean and $\Sigma$ variance,

$$\mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, \Sigma) \quad , \tag{4.6}$$

where

$$\Sigma = \begin{bmatrix} \sigma_\rho^2 & \sigma_{\rho\varphi} \\ \sigma_{\varphi\rho} & \sigma_\varphi^2 \end{bmatrix} \quad . \tag{4.7}$$

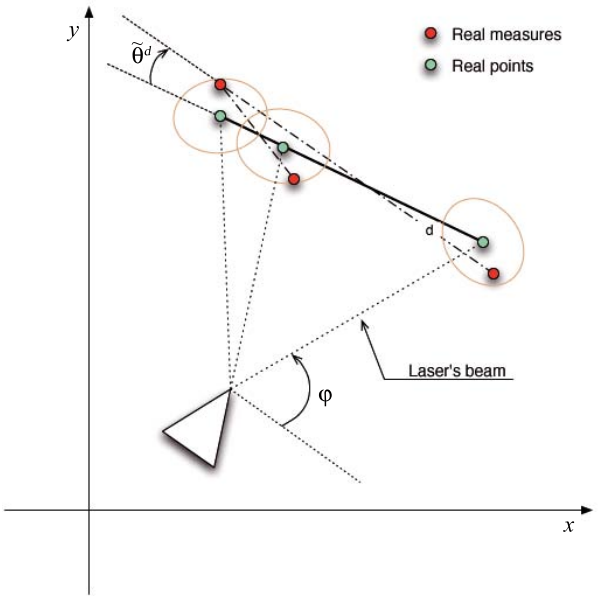



**Figure 4.3:** .

Also, we can assume that the error along the beam's direction $\rho$ is indipendent from the error along the angle $\varphi$, therefore $\sigma_{\rho\varphi} = \sigma_{\varphi\rho} = 0$. The probability density function ($pdf$) of $\mathbf{n}_t$ is

$$p(\rho, \varphi) = (2\pi)^{-1}|\det\Sigma|^{-1/2} \exp\Big\{ -\frac{1}{2} \begin{bmatrix}\rho & \varphi\end{bmatrix} \Sigma^{-1} \begin{bmatrix}\rho \\ \varphi\end{bmatrix} \Big\}, \tag{4.8}$$

We discuss here how to choose the distance $d_{min}$ between the two points (see Table 4.1). Of course, it's easy to understand that in order to minimize the transformation error due to the noise, we should set the distance to infinity. However, on the other hand, we would want to choose the points closed enough eachother, not only to guarantee to find them among the points of the next scans, but also to increase the probability $w$ to find a correct solution. In fact, greater the distance $d_{min}$ is, fewer points are feasible for finding the true transformation.

We want to minimize the error between the real rotation angle $\theta^*$ and the estimated rotation angle $\hat{\theta}$ (see fig. 4.3), that is

$$\min_d |\theta_i^* - \hat{\theta}_i| = \min_d |\tilde{\theta}_i| = \min_d |\tilde{\theta}_i^d + \tilde{\theta}_i^a| \quad , \quad i = 1, 2, 3, \ldots \tag{4.9}$$

where $\tilde{\theta}_i^d$ is the error due to the distance between the points selected for the estimation, $\tilde{\theta}_i^a$ is the error due to wrong associations and $d$ is the distance

between the two points selected for finding the transformation's parameters. The error $\tilde{\theta}_i^a$ does not depend from the distance between the two selected points. Therefore

$$\min_d |\tilde{\theta}_i| = \min_d |\tilde{\theta}_i^d| + \tilde{\theta}_i^a \quad , \quad i = 1, 2, 3, \ldots \tag{4.10}$$

So, what we need to minimize is the following

$$\min_d |\tilde{\theta}_i^d| \quad , \quad i = 1, 2, 3, \ldots \tag{4.11}$$

From the lidar technical data, the error variance along the beam direction is much greater than the error variance along the beam's angle. Thus, with good approximation, we can write

$$d \sin \tilde{\theta}^d \approx c\sigma_\rho \qquad \Longrightarrow \qquad \sin \tilde{\theta}^d \approx \frac{c}{d}\sigma_\rho \tag{4.12}$$

where $c$ is a constant that depends on the desired confidence.

For small values of $\tilde{\theta}^d$, we can use the $\sin \tilde{\theta}^d \simeq \tilde{\theta}^d$ approximation. Then, if we want a maximum error of $\varepsilon$, we have that

$$\tilde{\theta}^d \approx \frac{c}{d}\sigma_\rho \leq \varepsilon \tag{4.13}$$

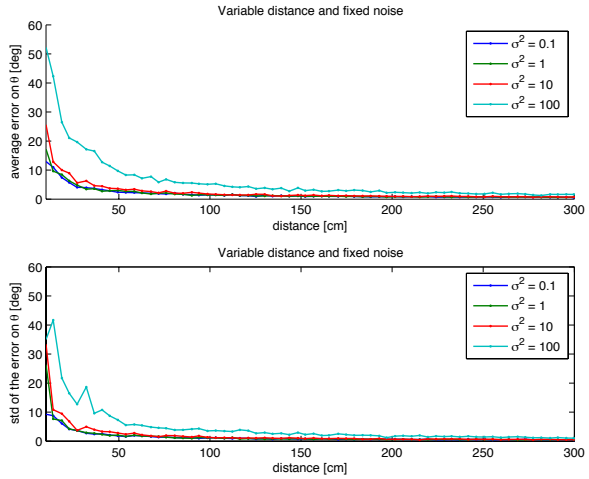
and therefore, in order to reduce the error due to the noise, the points should be choosen with a minimum retalive distance given by the following relation

$$d_{\mathcal{S}} \geq \frac{c}{\varepsilon}\sigma_\rho \tag{4.14}$$

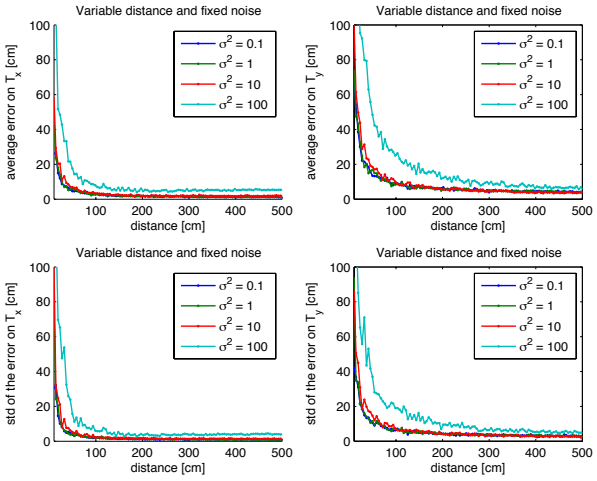
Experimental results are given in the Figure 4.4-4.5. In particular, in the Figure 4.4 it is shown the rotation and translation estimation errors increasing the distance between the two points. The bigger the distance is, the smaller the errors are. Notice also that the bigger the variance $\sigma_\rho^2$ is, the bigger the minimum distance for having an error under a certain threshold is, hence, if the fixed distance is too small, the resulting error is big even for small values of the noise variance (see Figure 4.5).

### 4.2.4 Simulation results

In this Section we show some visual results, using different data configurations. First, to test the validity of our approach, we perform a set of exact rigid registration experiments on some data sets without noise and outliers. Then, to test robustness and to see how our method behaves in presence of noise and outliers, we generate corrupted template point set from some model

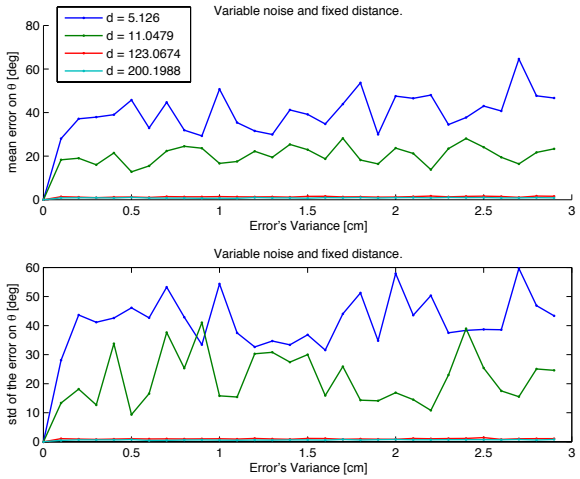



(a)

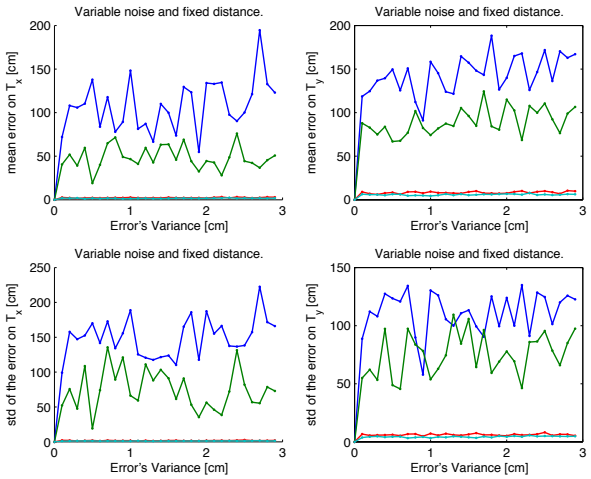



(b)

**Figure 4.4:** (a) Average error and standard deviation on the estimated rotation due to the distance among the selected control points, for different values of the noise strenght. (b) Average error and standard deviation on the estimated translation due to the distance among the selected control points, for different values of the noise strenght.




(a)

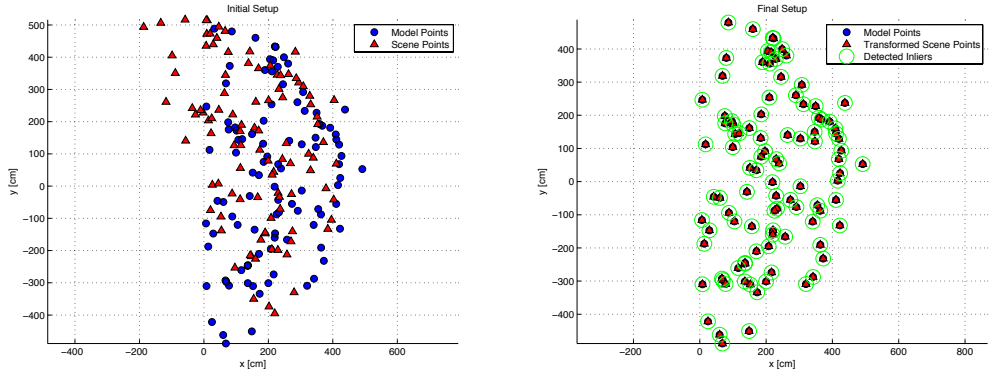



(b)

**Figure 4.5:** (a) Average error and standard deviation on the estimated rotation due to the distance among the selected control points, for different values of the distance $d_{min}$. (b) Average error and standard deviation on the estimated translation due to the distance among the selected control points, for different values of the distance $d_{min}$.

set. We register outlier corrupted point sets. We add outlier randomly drawn from a uniform distribution. We tested the algorithm increasing the outliers percentage with different configurations (rotations up to $\pi/4$ rad and translations up to 30 cm) of sets with 100 points each on a PowerBook G4 1.25 GHz, 512 MB Ram. The arrangement is the same as the previous one, i.e. on the left there is the model and superimposed unregistered scene; on the right there is the registered data after running our algorithm.

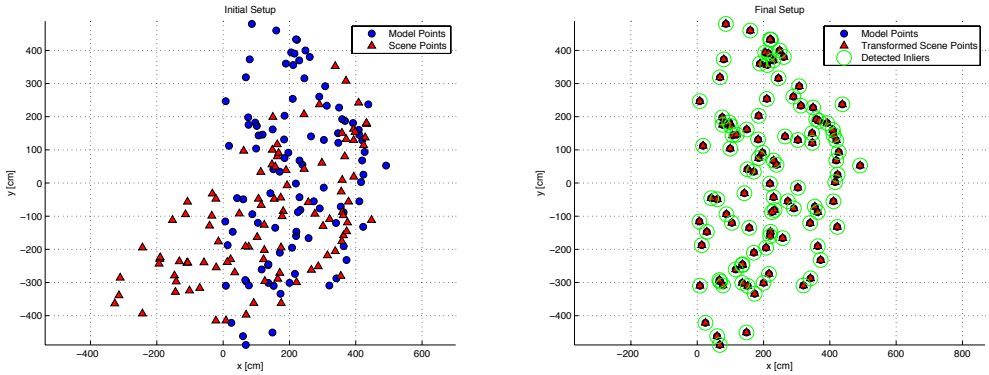
We begin to test the new algorithm over a data set without noise and outliers (Figure 4.6-4.7). The algorithm find the true parameters in a very




(a) (b)

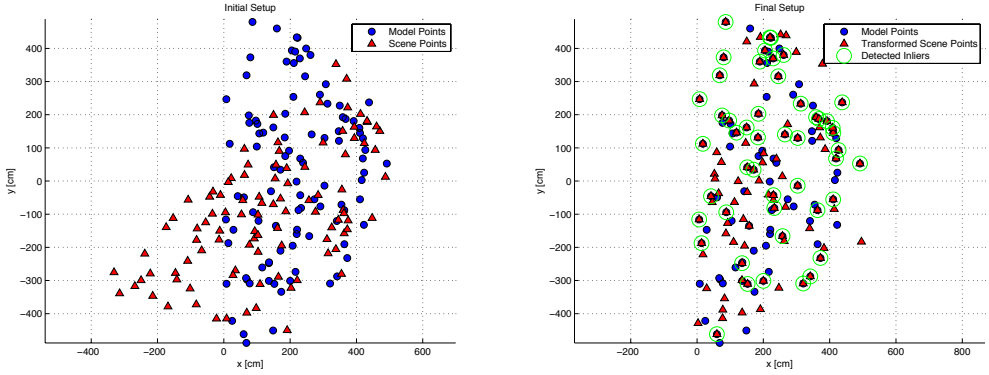
**Figure 4.6:** (a) Initial configuration: $N_m = N_s = 100$, $\theta = \pi/8$ $rad$, $T = [-30\ 30]^T$, no presence of noise and no outliers. (b) Registration result.

short execution time and it continues to work very well even with a large amount of outliers (Figure 4.8-4.9). Also ICPSAC is not sensible to the noise presence (Figure 4.10-4.11). Finally, the most important result is that its behavior is pretty good even in the case of noise presence with large outliers percentage (Figure 4.12-4.13).



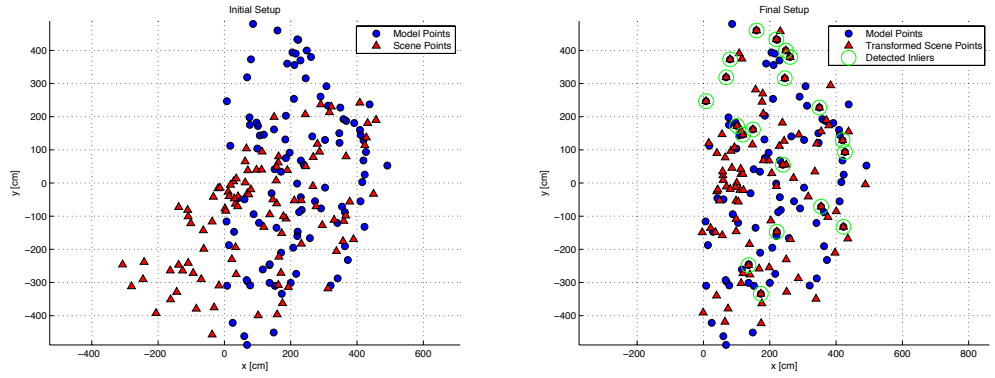

(a) (b)

**Figure 4.7:** (a) Initial configuration: $N_m = N_s = 100$, $\theta = -\pi/4\ rad$, $T = [-30\ 30]^T$, no presence of noise and no outliers. (b) Registration result.



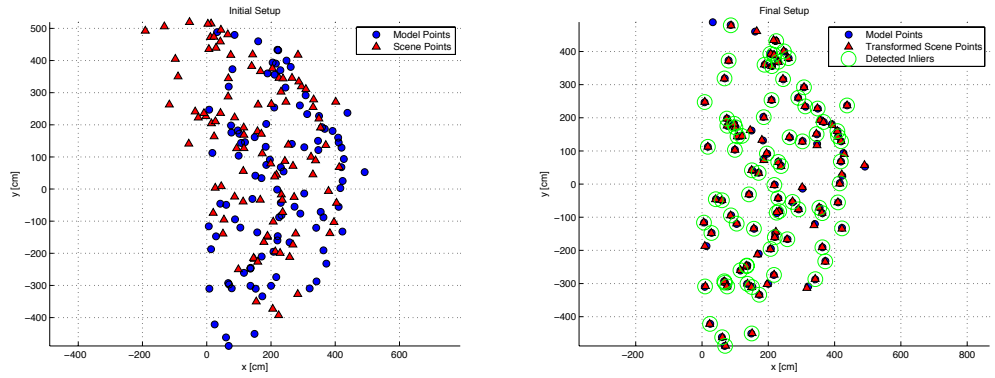

(a) (b)

**Figure 4.8:** (a) Initial configuration: $N_m = N_s = 100$, $\theta = -\pi/4\ rad$, $T = [-30\ 30]^T$, no presence of noise and 50% of outliers. (b) Registration result.



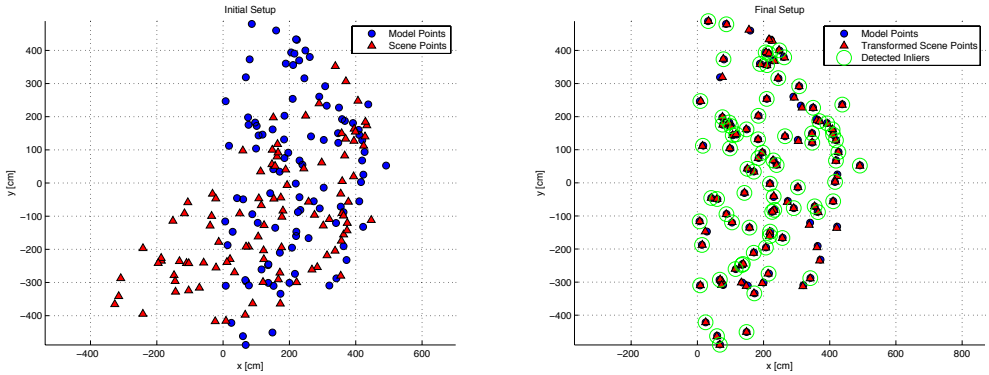

(a) (b)

**Figure 4.9:** (a) Initial configuration: $N_m = N_s = 100$, $\theta = -\pi/4$ *rad*, $T = [-30\ 30]^T$, no presence of noise and 80% of outliers. (b) Registration result.



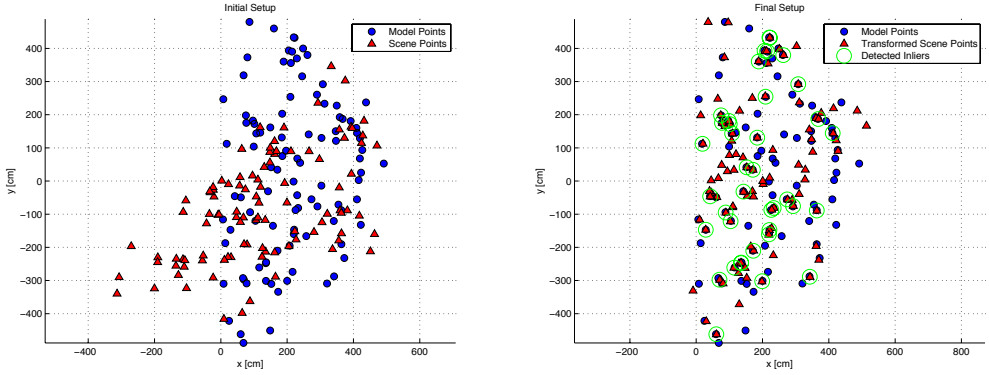

(a) (b)

**Figure 4.10:** (a) Initial configuration: $N_m = N_s = 100$, $\theta = \pi/8$ *rad*, $T = [-30\ 30]^T$, presence of noise and no outliers. (b) Registration result.



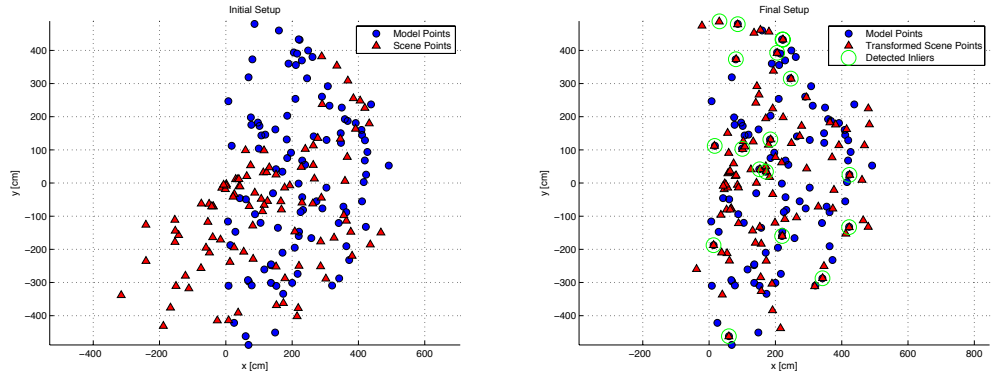

(a) (b)

**Figure 4.11:** (a) Initial configuration: $N_m = N_s = 100$, $\theta = -\pi/4\ rad$, $T = [-30\ 30]^T$, presence of noise and no outliers. (b) Registration result.




(a) (b)

**Figure 4.12:** (a) Initial configuration: $N_m = N_s = 100$, $\theta = -\pi/4\ rad$, $T = [-30\ 30]^T$, presence of noise and 50% of outliers. (b) Registration result.




(a) (b)

**Figure 4.13:** (a) Initial configuration: $N_m = N_s = 100$, $\theta = -\pi/4\ rad$, $T = [-30\ 30]^T$, presence of noise and 80% of outliers. (b) Registration result.

### 4.2.5 Simulated overall results

In the following figures we show the ICPSAC's behaviour for various parameters values, both for the noiseless case that the noisy case. Finally, we give the overall results obtained with the best paramenter's values choice for different rotation angle values: the noiseless case in the Figure 4.26-4.29 and the noisy case in the Figure 4.42-4.45. From the figures, one can see that the average rotation error is always bounded and less than $0.01°$, while the average translation error is always bounded as well and less than 0.05 cm, even with 80% of outliers. Even the probability error, defined as the number of times the algorithm is not able to find a correct registration, is different from zero only with the 80% of outliers for the reason that in some cases the inliers intersection is void.

We can conclude saying that ICPSAC is stable not only in presence of noise, in fact there are not remarkable differences between the noiseless case and the noisy one, but also respect to an increasing outliers percentage. It works pretty well even with the 80% of outliers, whenever a possible solution may be found.

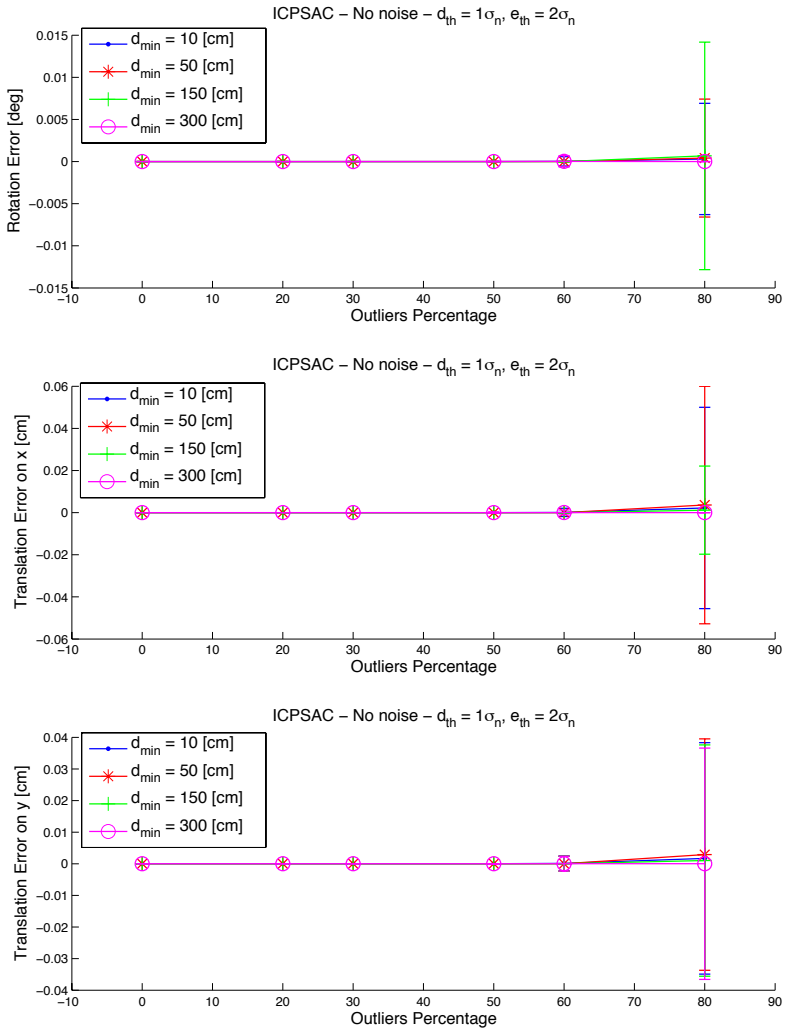



**Figure 4.14:** Average registration error and standard deviation, respectively on the angle rotation $\theta$ and the translations $T_x$ and $T_y$, obtained by using the ICPSAC algorithm without noise for different values of the parameter $d_{min}$.

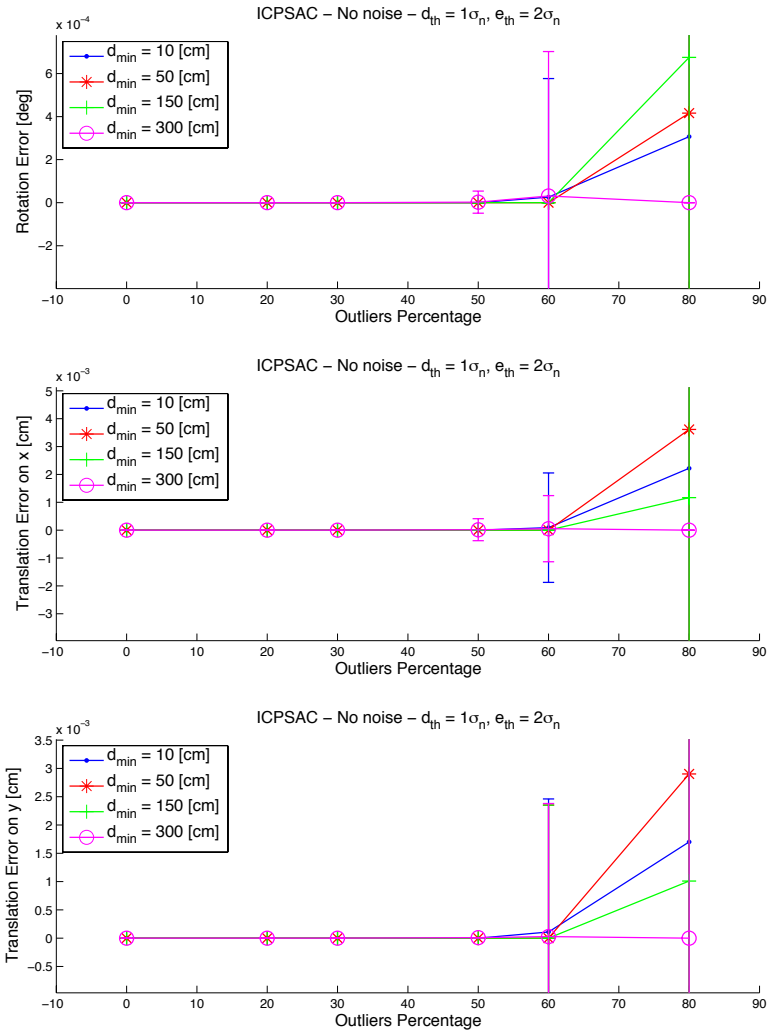



**Figure 4.15:** Average registration error and standard deviation, respectively on the angle rotation $\theta$ and the translations $T_x$ and $T_y$, obtained by using the ICPSAC algorithm without noise for different values of the parameter $d_{min}$ (enlargement).

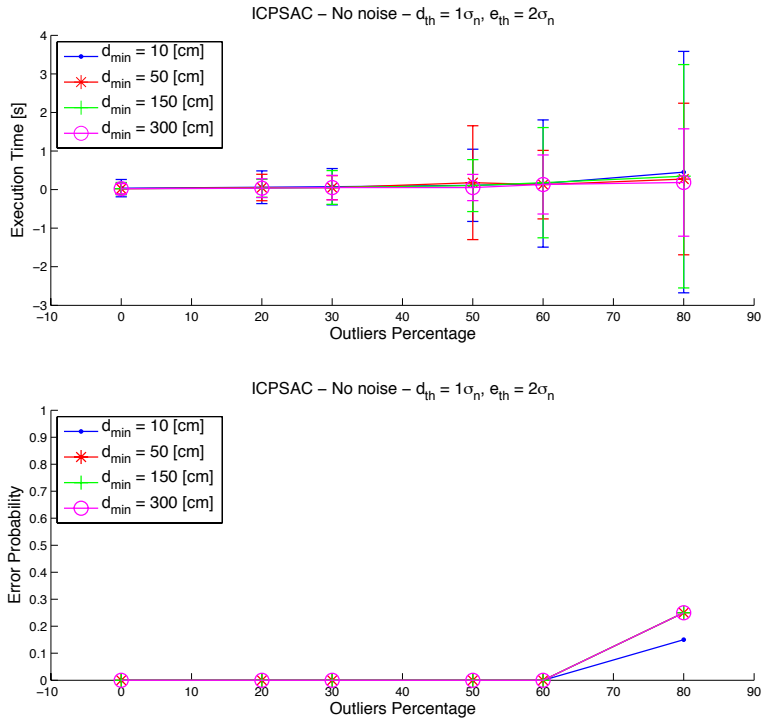



**Figure 4.16:** Execution time and error probability of the ICPSAC algorithm without noise for different values of the parameter $d_{min}$.

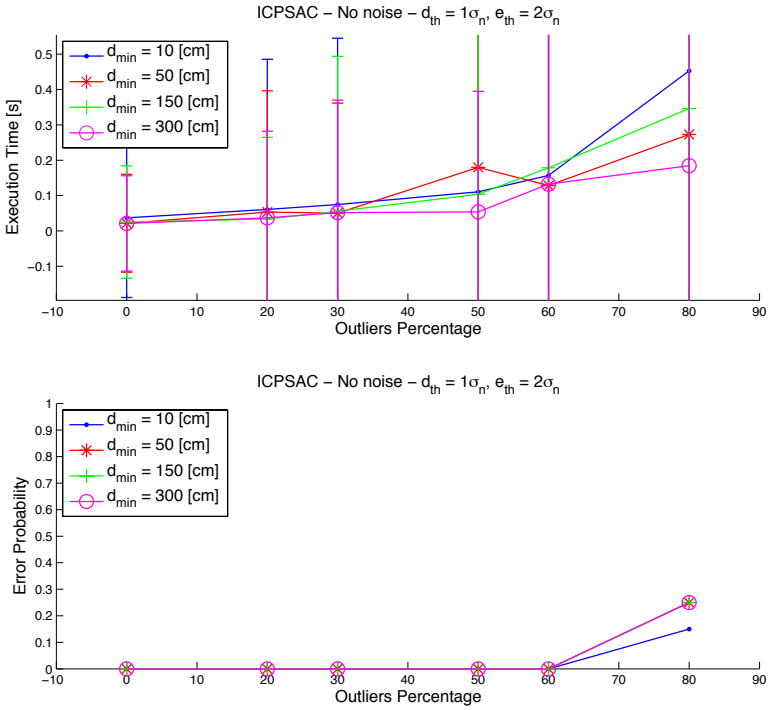



**Figure 4.17:** Execution time and error probability of the ICPSAC algorithm without noise for different values of the parameter $d_{min}$ (enlargement).

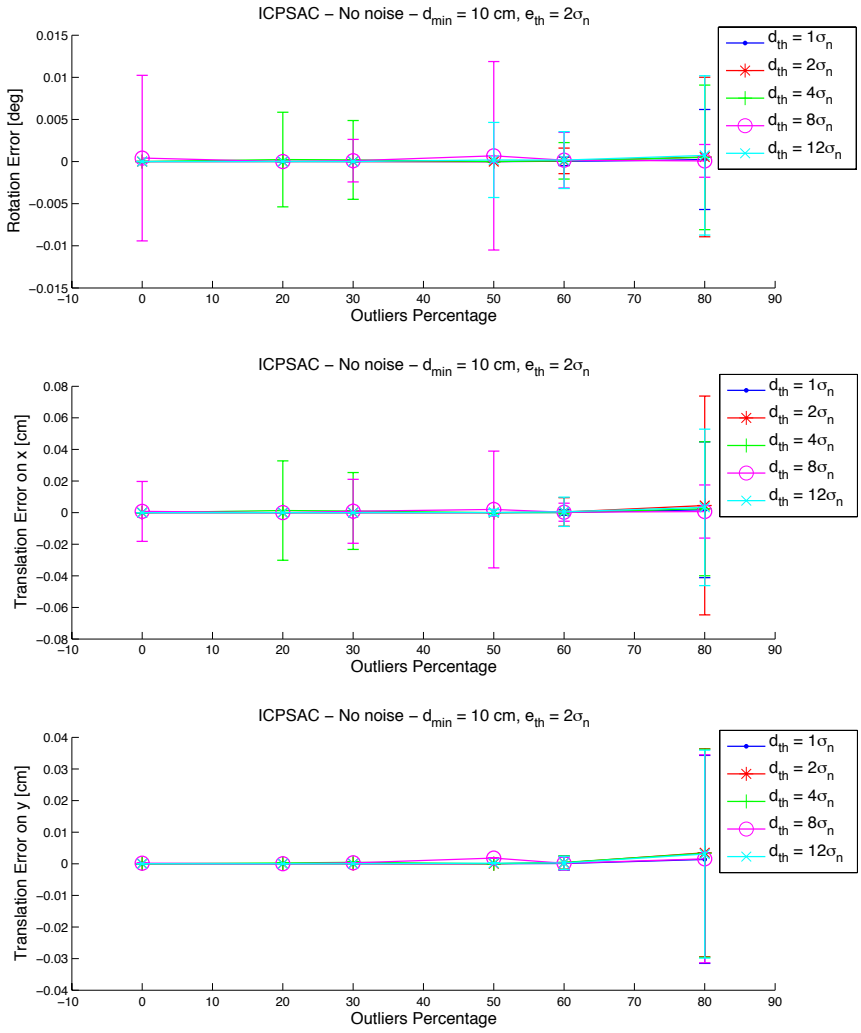



**Figure 4.18:** Average registration error and standard deviation, respectively on the angle rotation $\theta$ and the translations $T_x$ and $T_y$, obtained by using the ICPSAC algorithm without noise for different values of the parameter $d_{th}$.

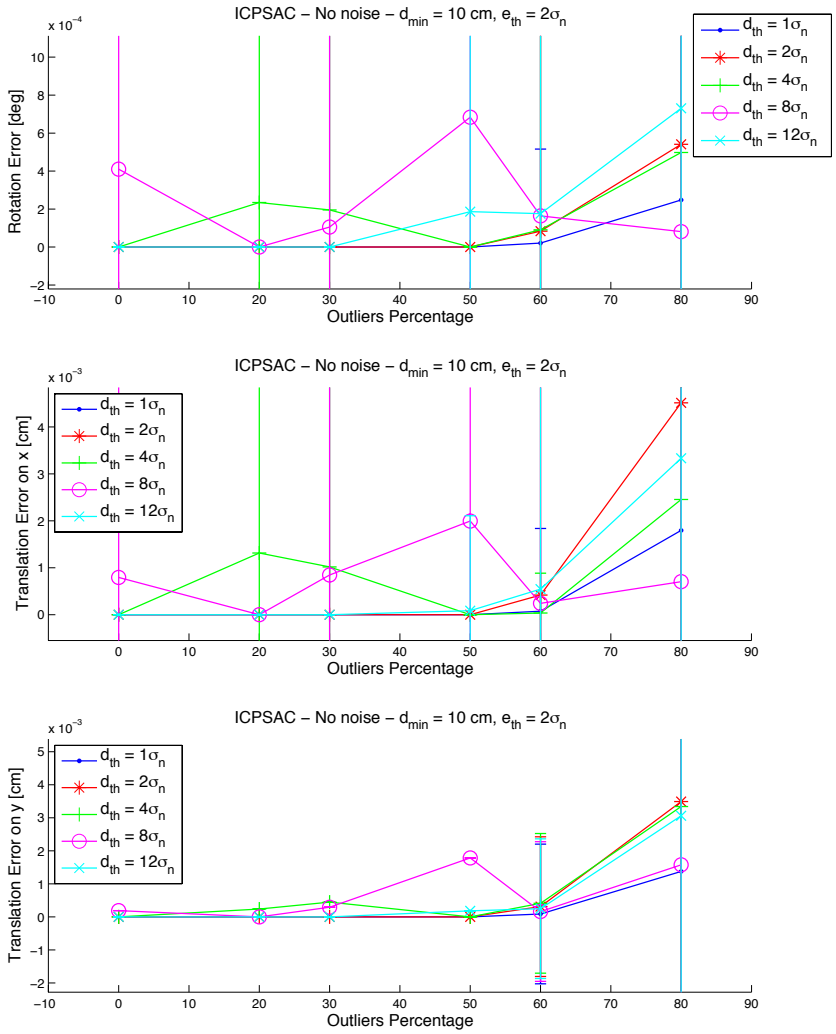



**Figure 4.19:** Average registration error and standard deviation, respectively on the angle rotation $\theta$ and the translations $T_x$ and $T_y$, obtained by using the ICPSAC algorithm without noise for different values of the parameter $d_{th}$ (enlargement).

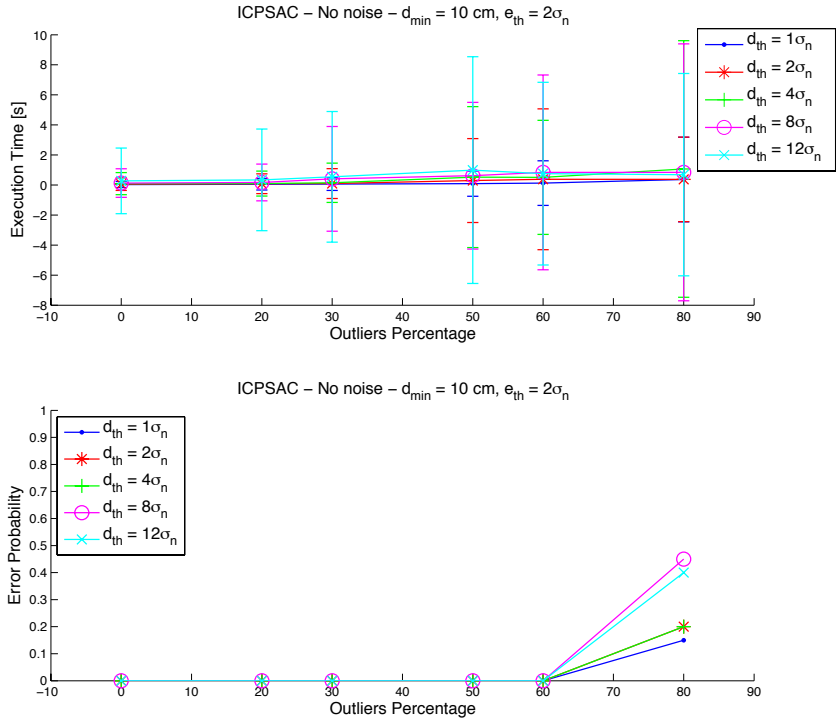



**Figure 4.20:** Execution time and error probability of the ICPSAC algorithm without noise for different values of the parameter $d_{th}$.

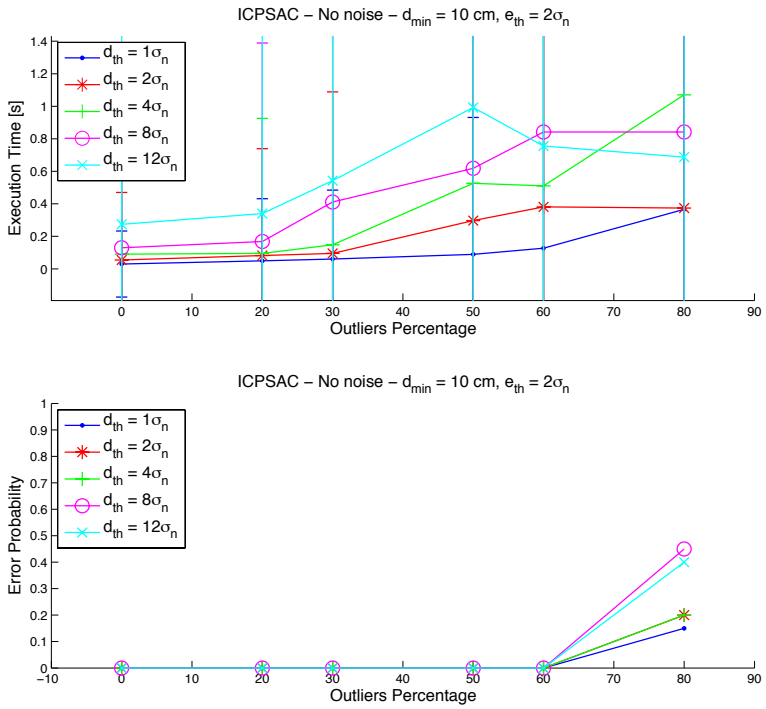



**Figure 4.21:** Execution time and error probability of the ICPSAC algorithm without noise for different values of the parameter $d_{th}$ (enlargement).

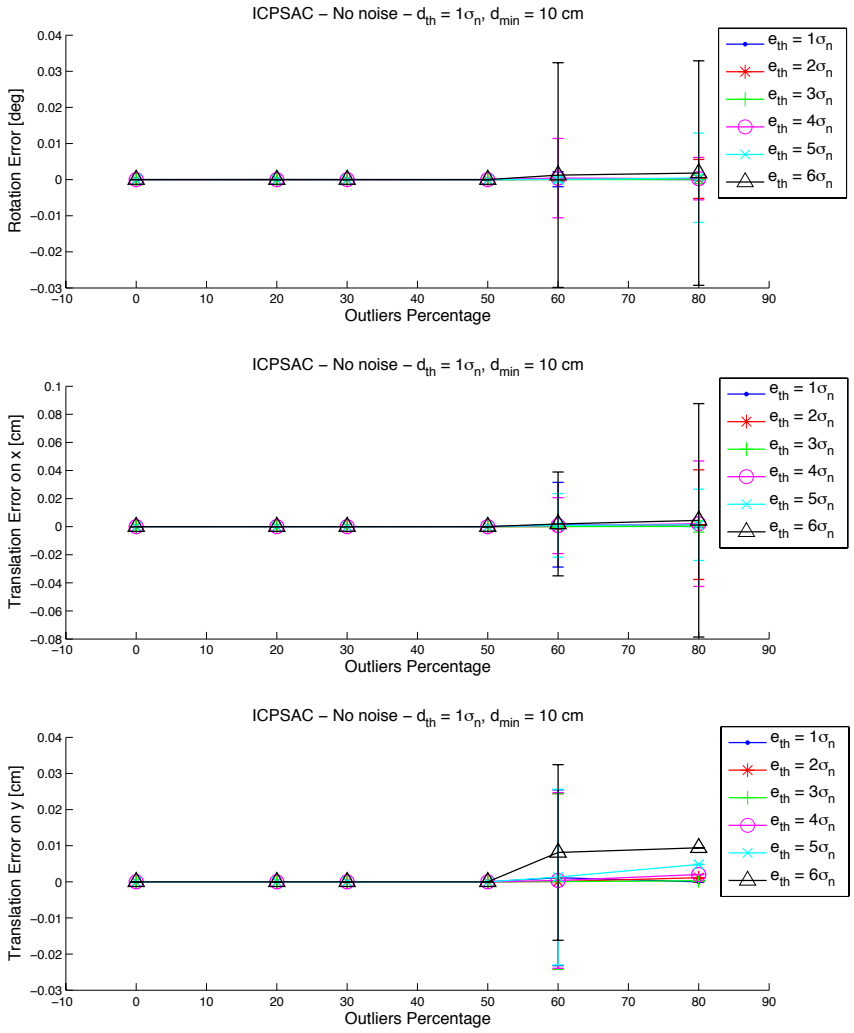



**Figure 4.22:** Average registration error and standard deviation, respectively on the angle rotation $\theta$ and the translations $T_x$ and $T_y$, obtained by using the ICPSAC algorithm without noise for different values of the parameter $e_{th}$.

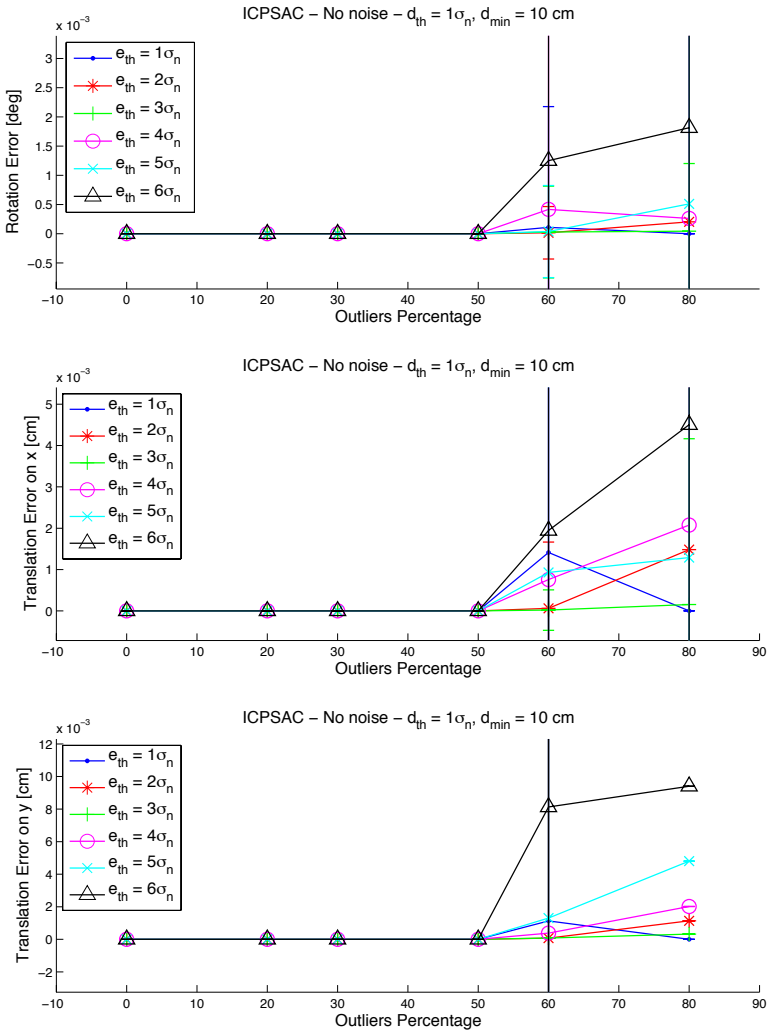



**Figure 4.23:** Average registration error and standard deviation, respectively on the angle rotation $\theta$ and the translations $T_x$ and $T_y$, obtained by using the ICPSAC algorithm without noise for different values of the parameter $e_{th}$ (enlargement).

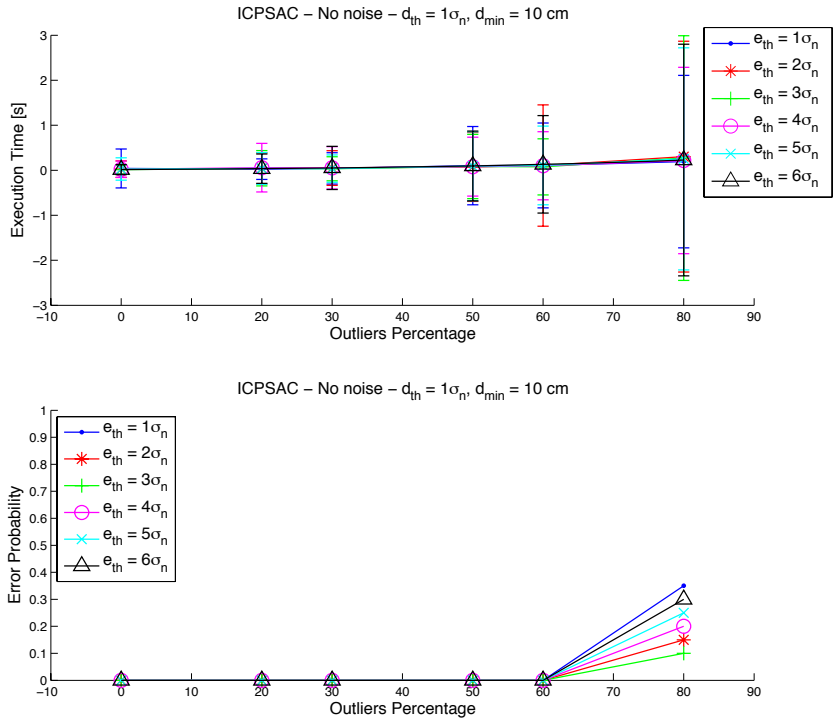



**Figure 4.24:** Execution time and error probability of the ICPSAC algorithm without noise for different values of the parameter $e_{th}$.

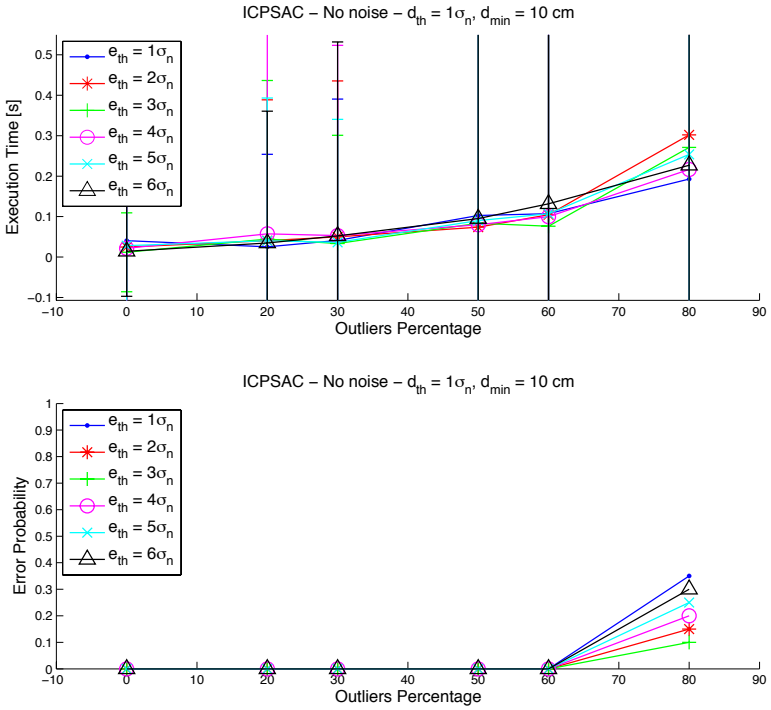



**Figure 4.25:** Execution time and error probability of the ICPSAC algorithm without noise for different values of the parameter $e_{th}$ (enlargement).

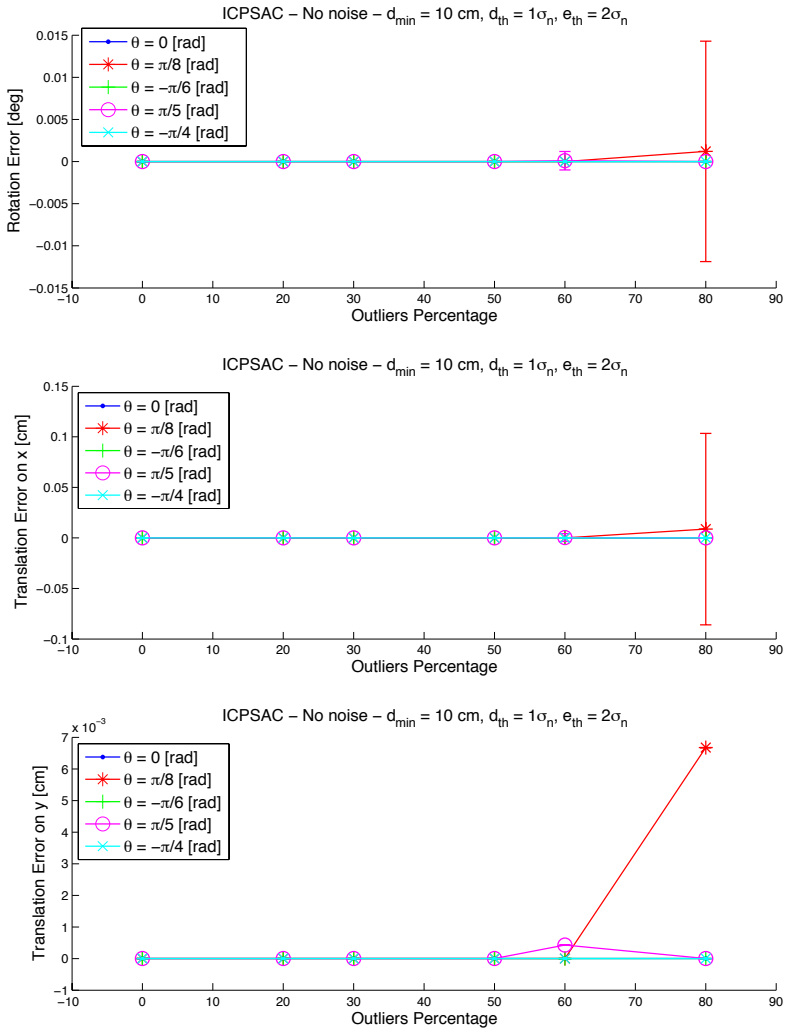



**Figure 4.26:** Average registration error and standard deviation, respectively on the angle rotation $\theta$ and the translations $T_x$ and $T_y$, obtained by using the ICPSAC algorithm without noise for different values of the angle rotation $\theta$.

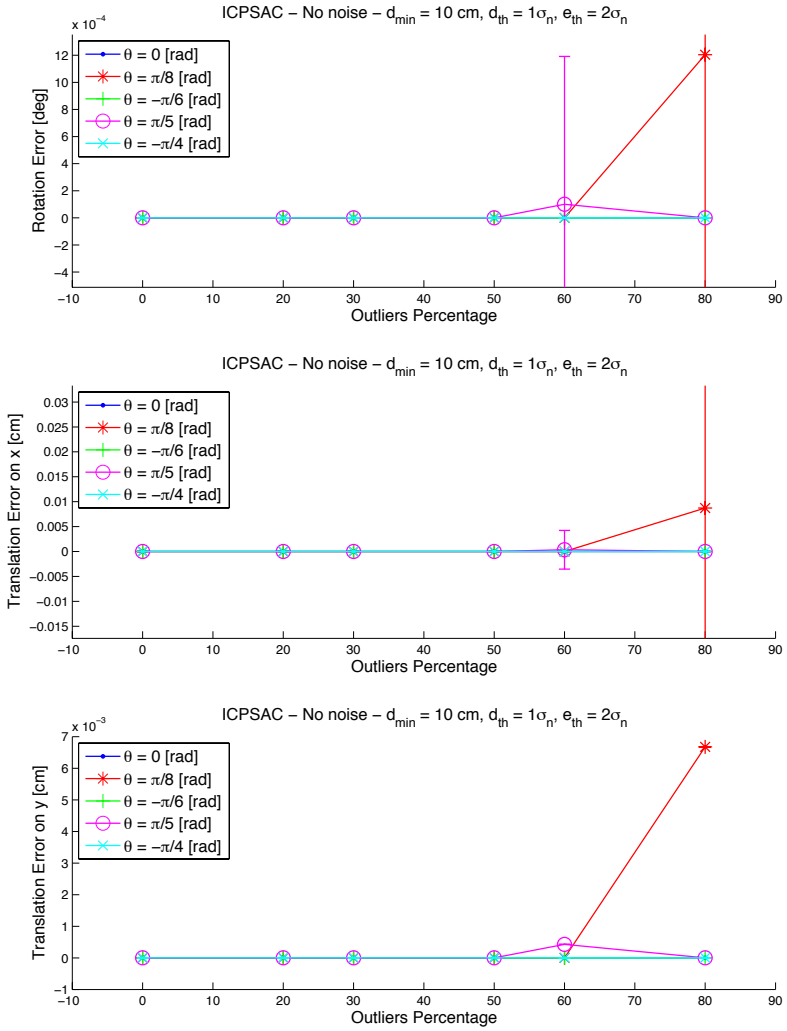



**Figure 4.27:** Average registration error and standard deviation, respectively on the angle rotation $\theta$ and the translations $T_x$ and $T_y$, obtained by using the ICPSAC algorithm without noise for different values of the angle rotation $\theta$ (enlargement).

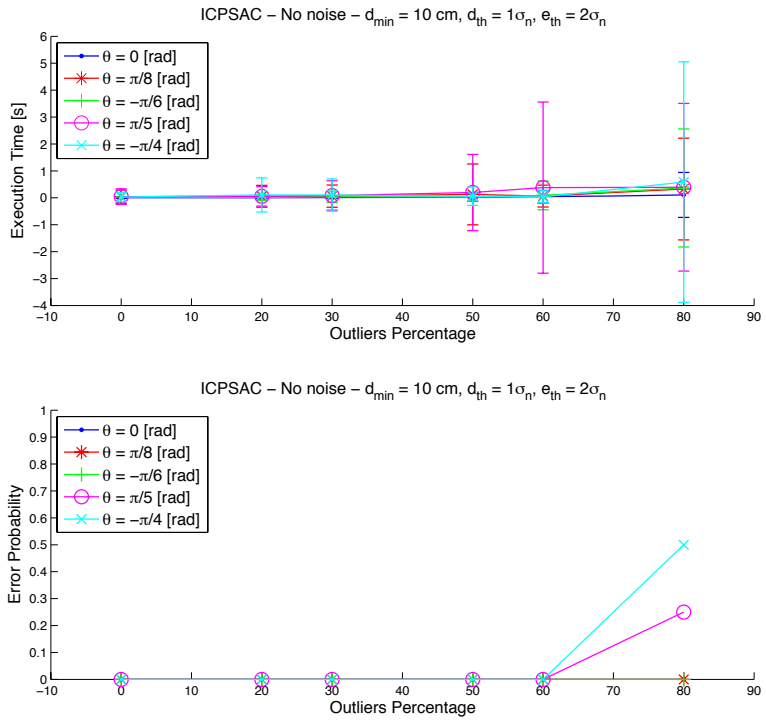



**Figure 4.28:** Execution time and error probability of the ICPSAC algorithm without noise for different values of the angle rotation $\theta$.

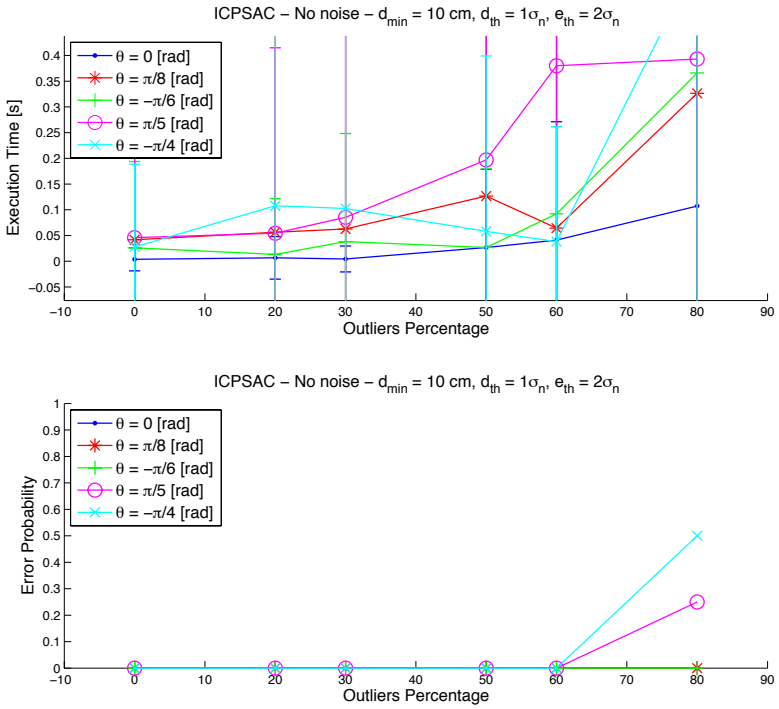



**Figure 4.29:** Execution time and error probability of the ICPSAC algorithm without noise for different values of the angle rotation $\theta$ (enlargement).

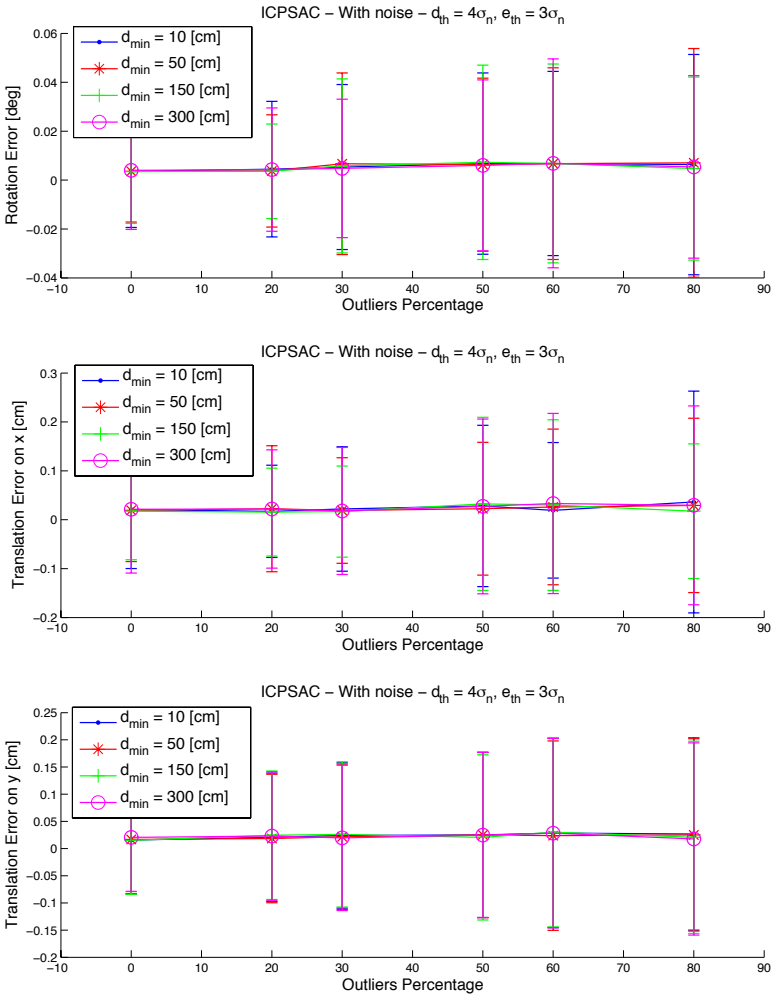



**Figure 4.30:** Average registration error and standard deviation, respectively on the angle rotation $\theta$ and the translations $T_x$ and $T_y$, obtained by using the ICPSAC algorithm with noise for different values of the parameter $d_{min}$.

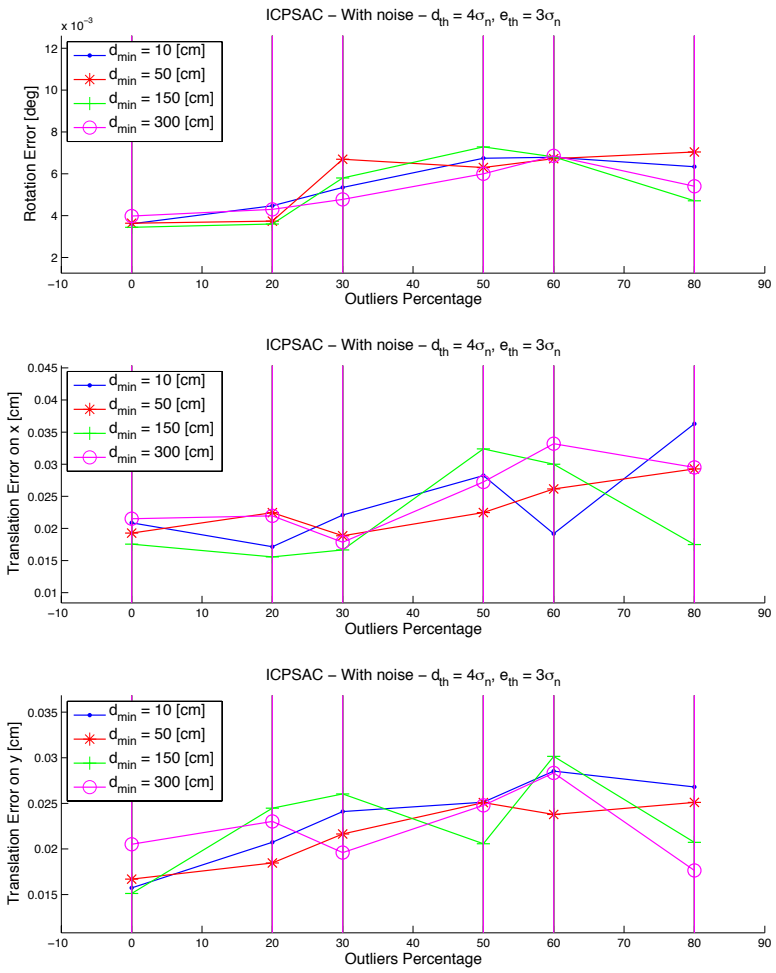



**Figure 4.31:** Average registration error and standard deviation, respectively on the angle rotation $\theta$ and the translations $T_x$ and $T_y$, obtained by using the ICPSAC algorithm with noise for different values of the parameter $d_{min}$ (enlargement).

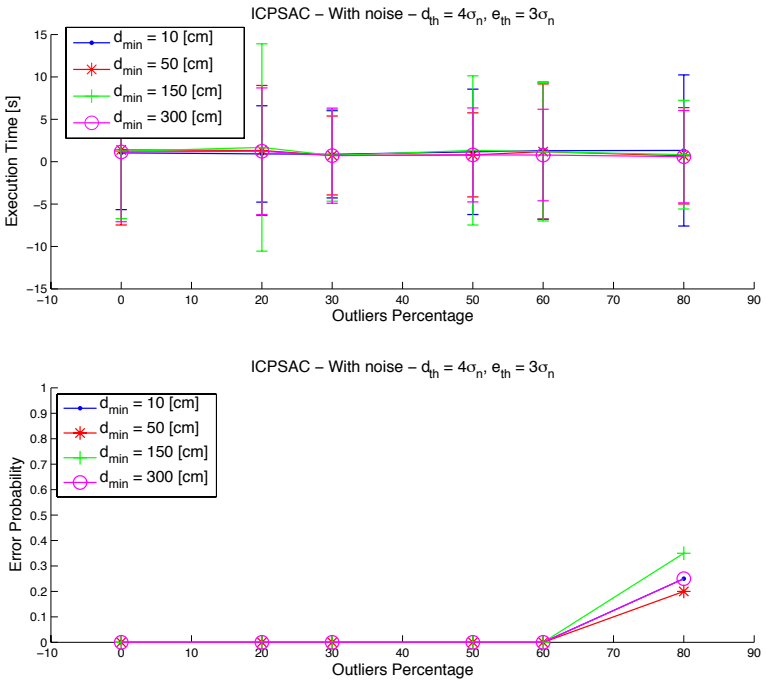



**Figure 4.32:** Execution time and error probability of the ICPSAC algorithm with noise for different values of the parameter $d_{min}$.

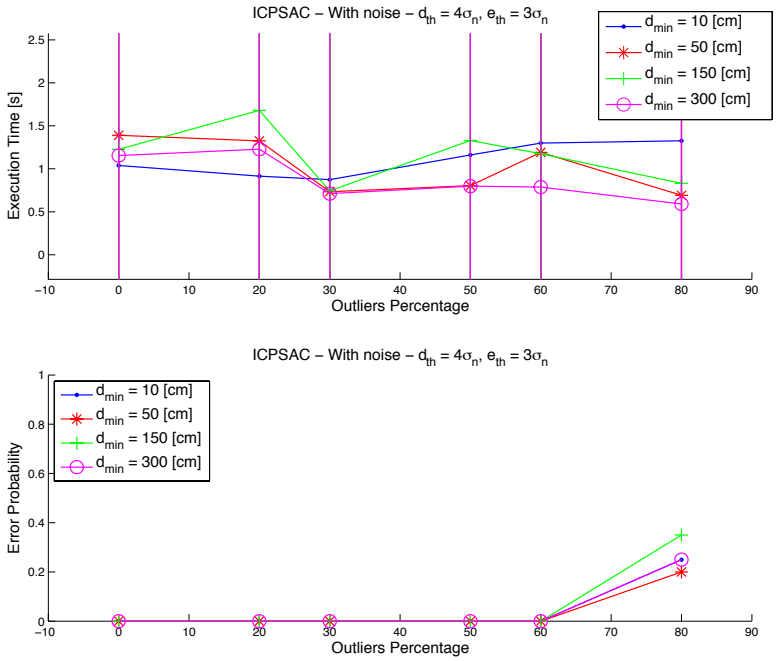



**Figure 4.33:** Execution time and error probability of the ICPSAC algorithm with noise for different values of the parameter $d_{min}$ (enlargement).

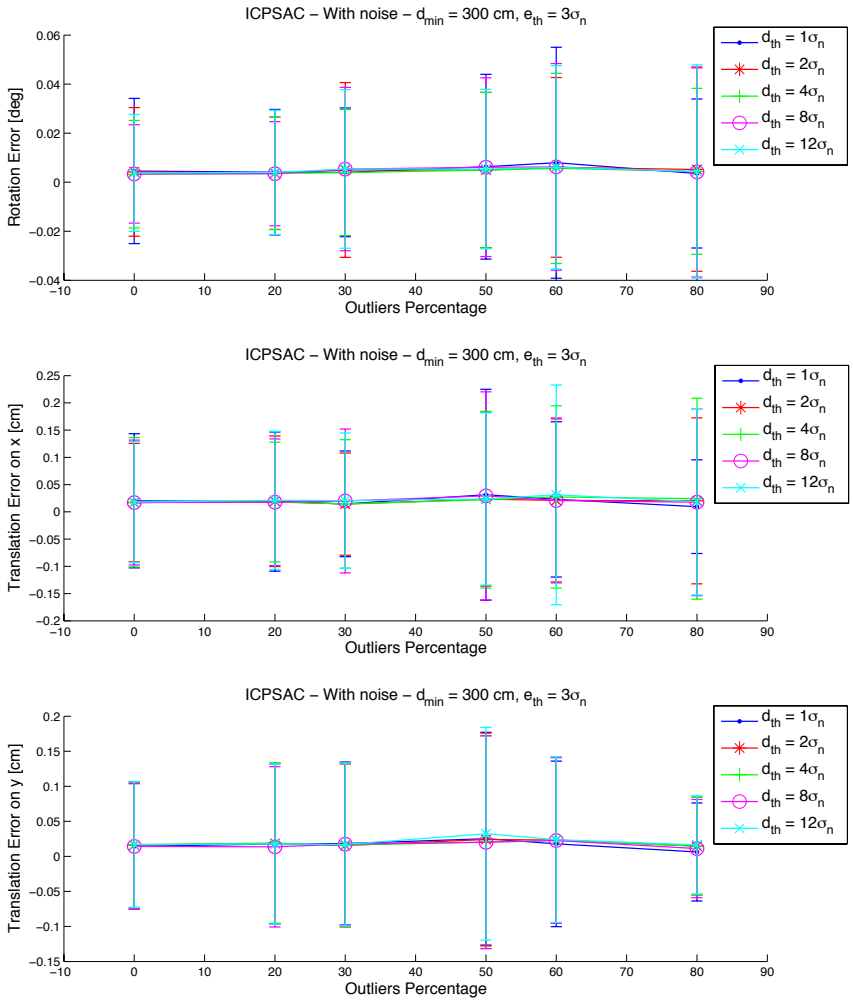



**Figure 4.34:** Average registration error and standard deviation, respectively on the angle rotation $\theta$ and the translations $T_x$ and $T_y$, obtained by using the ICPSAC algorithm with noise for different values of the parameter $d_{th}$.

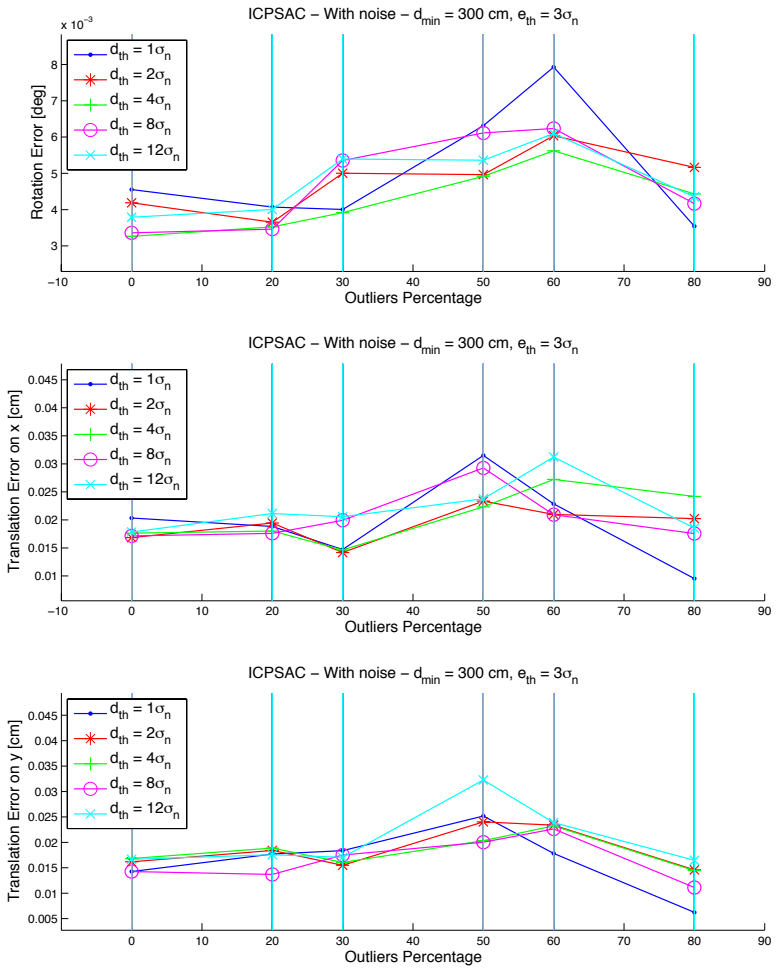



**Figure 4.35:** Average registration error and standard deviation, respectively on the angle rotation $\theta$ and the translations $T_x$ and $T_y$, obtained by using the ICPSAC algorithm with noise for different values of the parameter $d_{th}$ (enlargement).

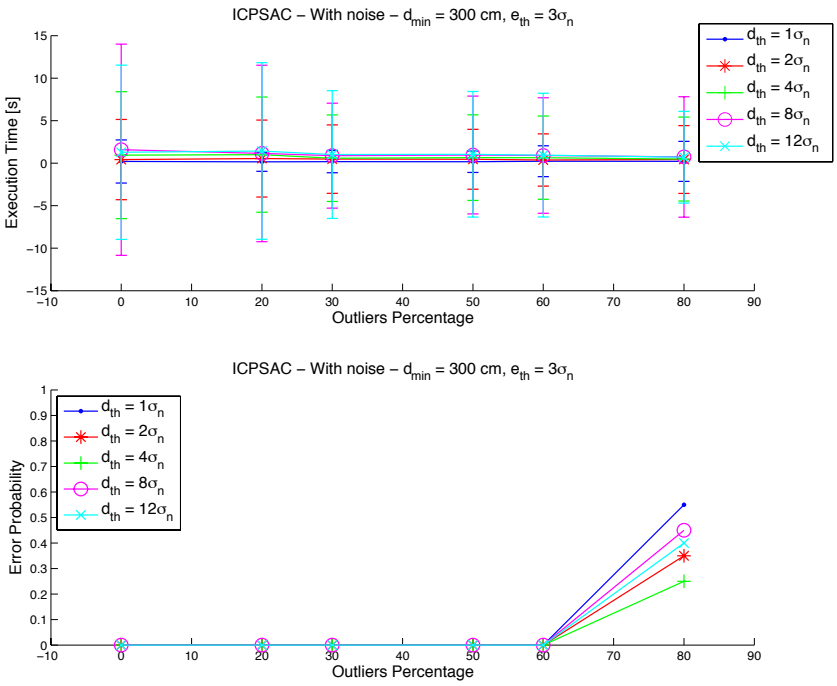



**Figure 4.36:** Execution time and error probability of the ICPSAC algorithm with noise for different values of the parameter $d_{th}$.

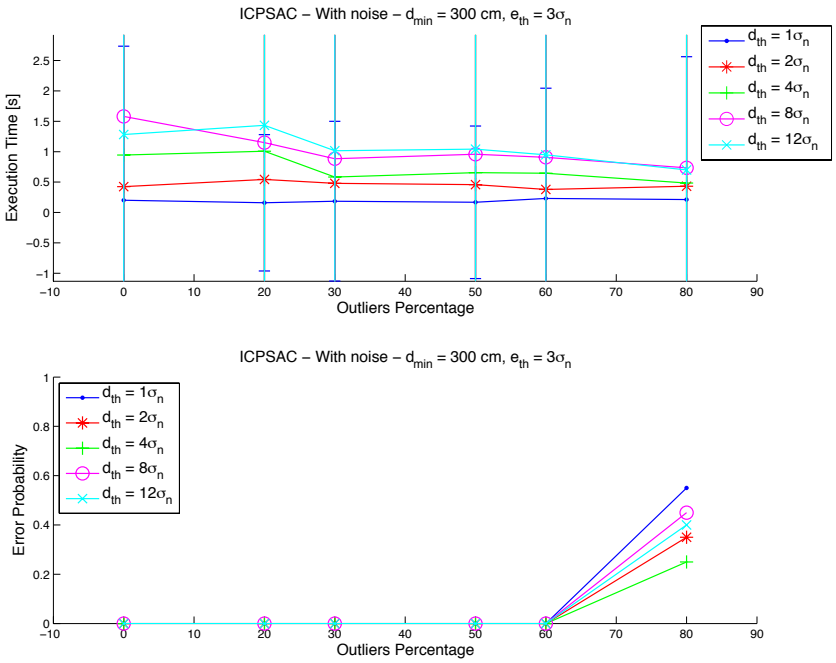



**Figure 4.37:** Execution time and error probability of the ICPSAC algorithm with noise for different values of the parameter $d_{th}$ (enlargement).

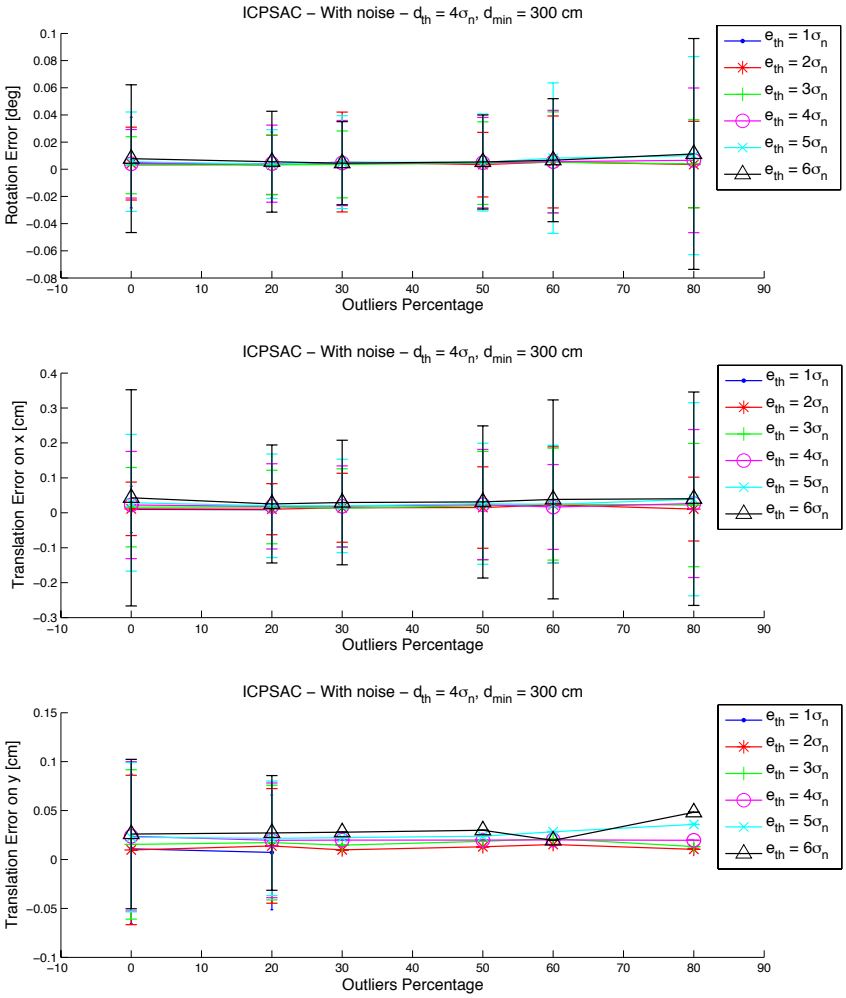



**Figure 4.38:** Average registration error and standard deviation (only the possible values are shown), respectively on the angle rotation $\theta$ and the translations $T_x$ and $T_y$, obtained by using the ICPSAC algorithm with noise for different values of the parameter $e_{th}$.

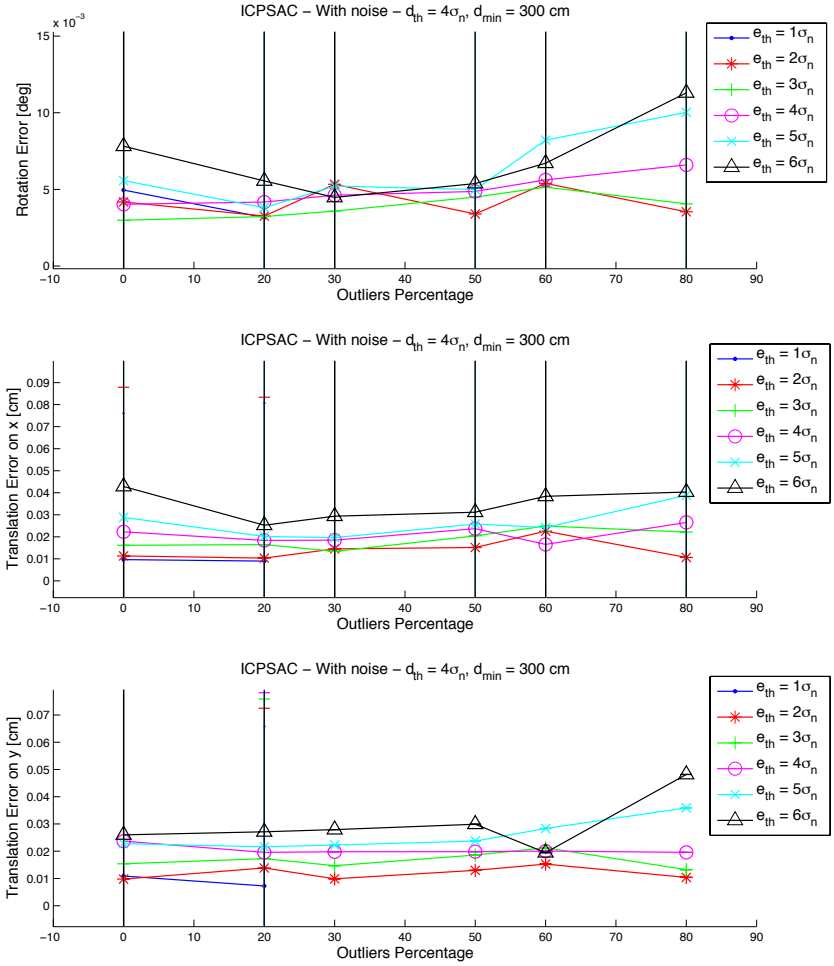



**Figure 4.39:** Average registration error and standard deviation (only the possible values are shown), respectively on the angle rotation $\theta$ and the translations $T_x$ and $T_y$, obtained by using the ICPSAC algorithm with noise for different values of the parameter $e_{th}$ (enlargement).

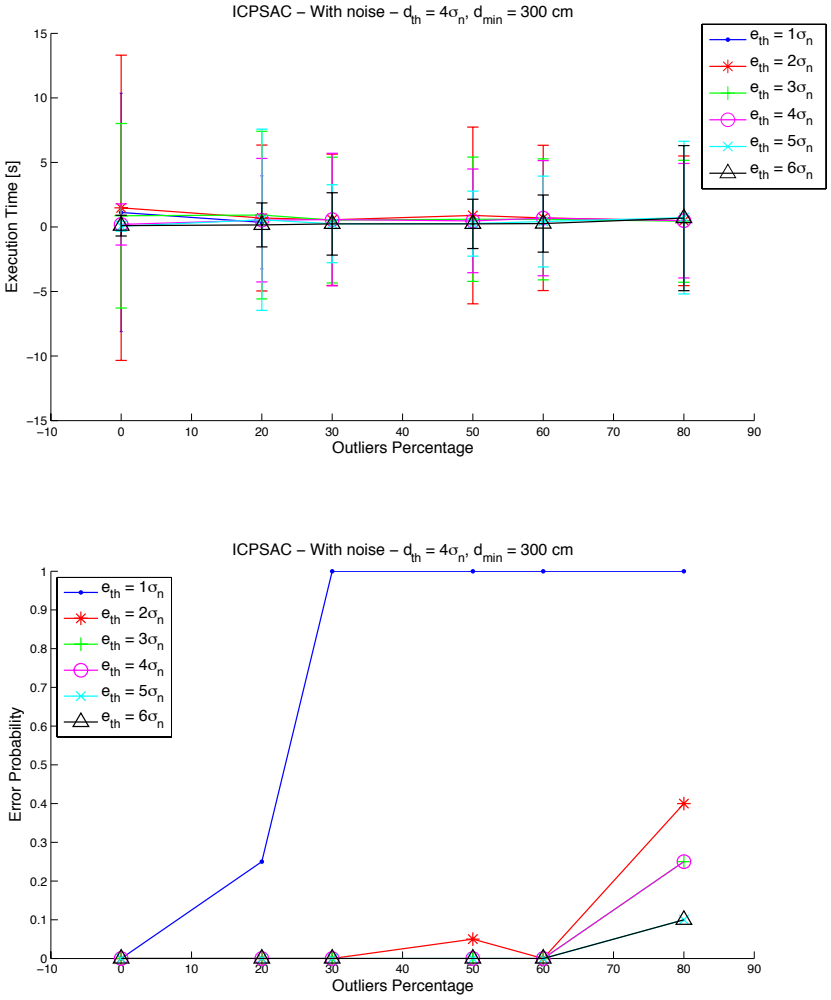



**Figure 4.40:** Execution time and error probability (only the possible values are shown) of the ICPSAC algorithm with noise for different values of the parameter $e_{th}$.

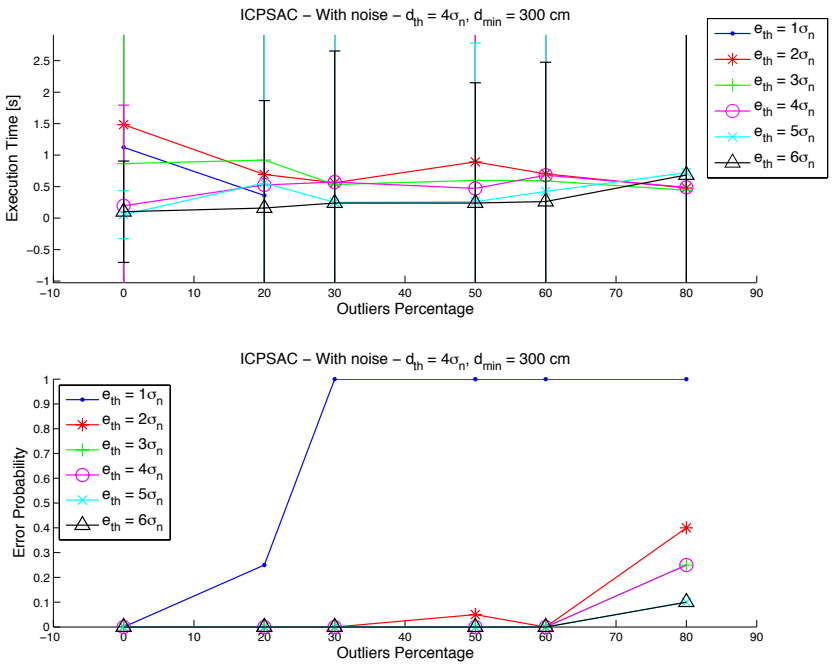



**Figure 4.41:** Execution time and error probability (only the possible values are shown) of the ICPSAC algorithm with noise for different values of the parameter $e_{th}$ (enlargement).

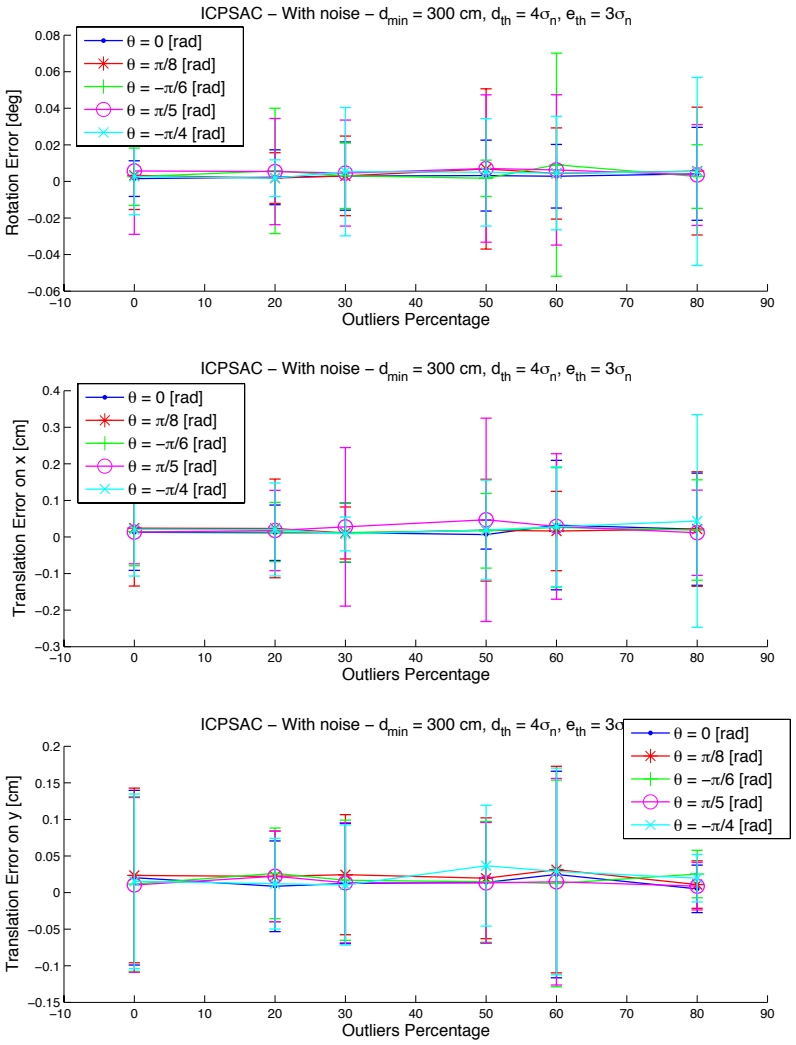



**Figure 4.42:** Average registration error and standard deviation, respectively on the angle rotation $\theta$ and the translations $T_x$ and $T_y$, obtained by using the ICPSAC algorithm with noise for different values of the angle rotation $\theta$ (only the possible values are shown).

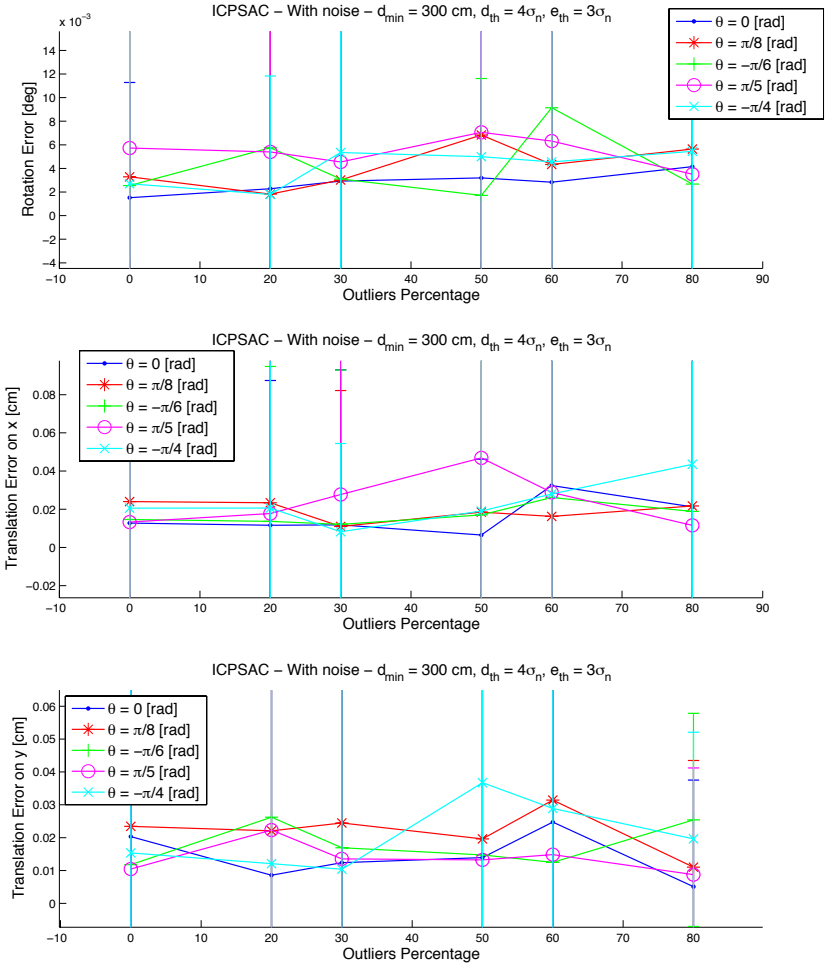



**Figure 4.43:** Average registration error and standard deviation, respectively on the angle rotation $\theta$ and the translations $T_x$ and $T_y$, obtained by using the ICPSAC algorithm with noise for different values of the angle rotation $\theta$ (enlargement).

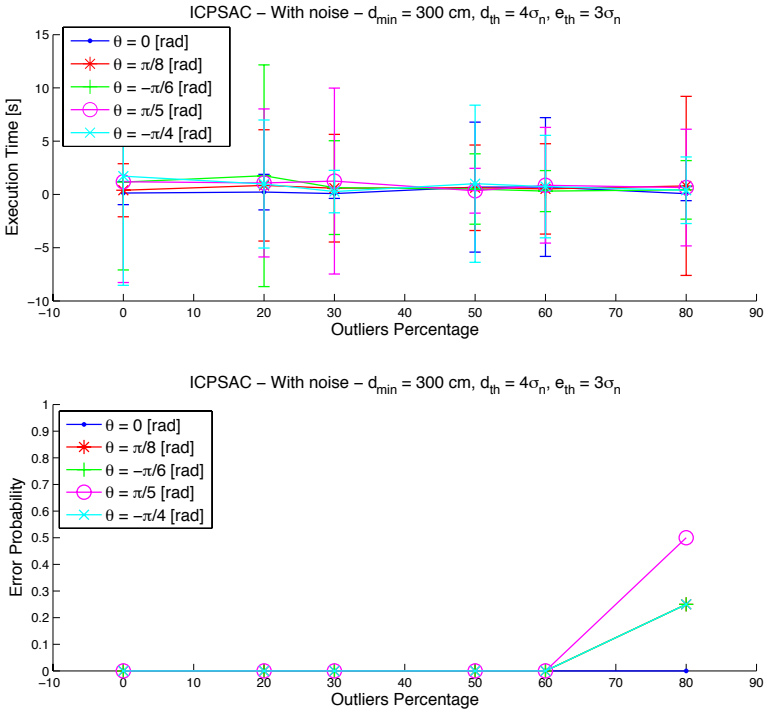



**Figure 4.44:** Execution time and error probability of the ICPSAC algorithm with noise for different values of the angle rotation $\theta$.

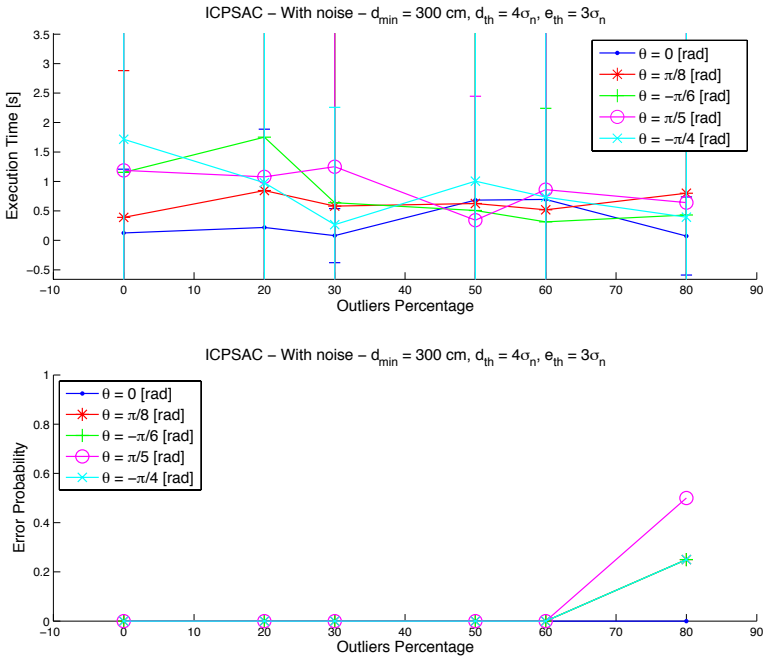
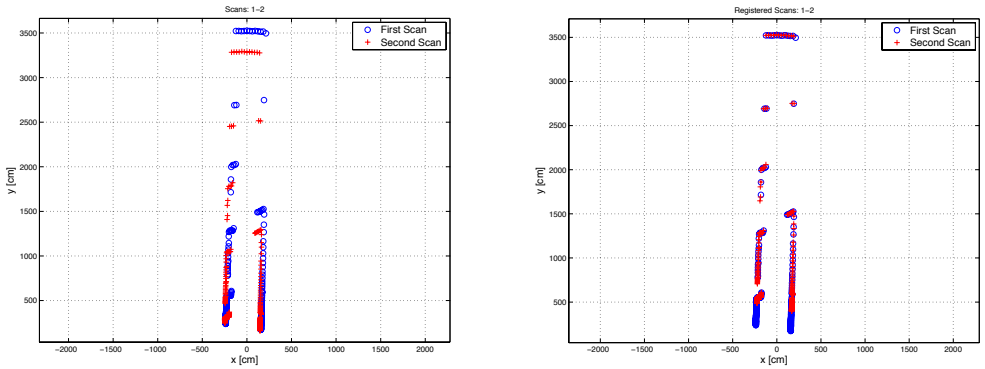



**Figure 4.45:** Execution time and error probability of the ICPSAC algorithm with noise for different values of the angle rotation $\theta$ (enlargement).

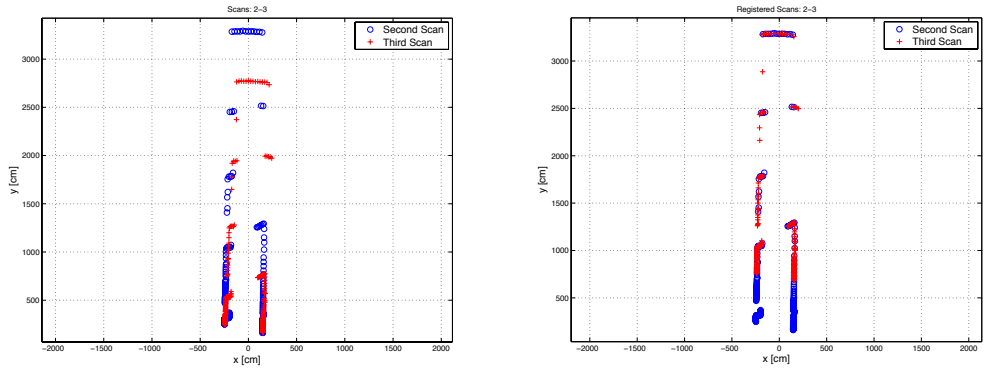
### 4.2.6 Experimental results

We show in this section some experimental results. We took five measurement scans along a building's corridor with the LMS 291 mounted on a robot chassis. In the Figure 4.46-4.49 are shown the single registration results and finally the Figure 4.50 shows the overall result of the map reconstrunction and of the robot pose estimation using five sequential scans. The mobile robot can reliably construct a good map of the environment (here, the third floor of Computer Science Department at the UCLA) while simultaneously localizing itself and this is accomplished using only a scanning sensor, the lidar.



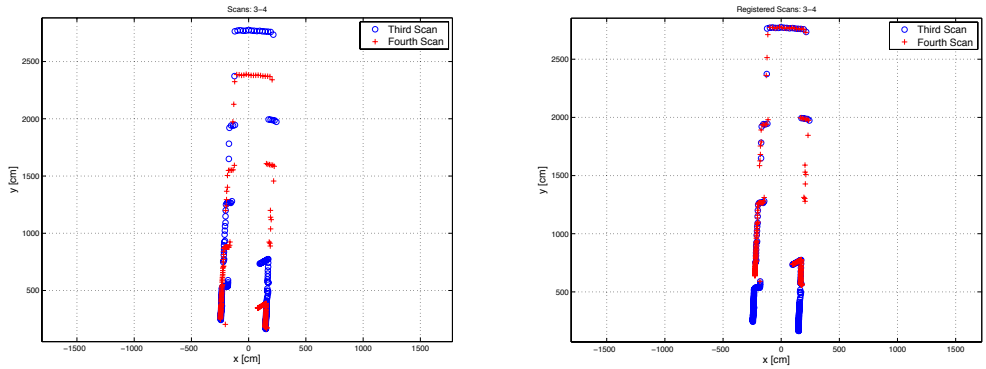

(a) (b)

**Figure 4.46:** Experimental results: (a) first and second lidar's scans; (b) registration result.



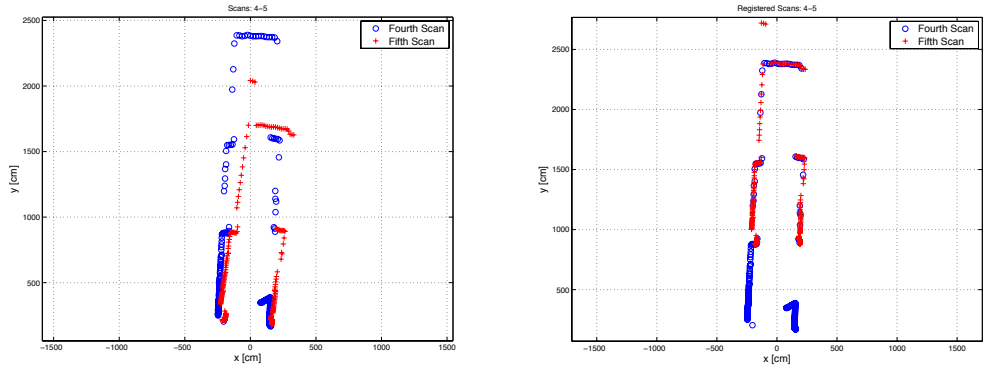

(a) (b)

**Figure 4.47:** Experimental results: (a) second and third lidar's scans; (b) registration result.




(a) (b)

**Figure 4.48:** Experimental results: (a) third and fourth lidar's scans; (b) registration result.




(a) (b)

**Figure 4.49:** Experimental results: (a) fourth and fifth lidar's scans; (b) registration result.

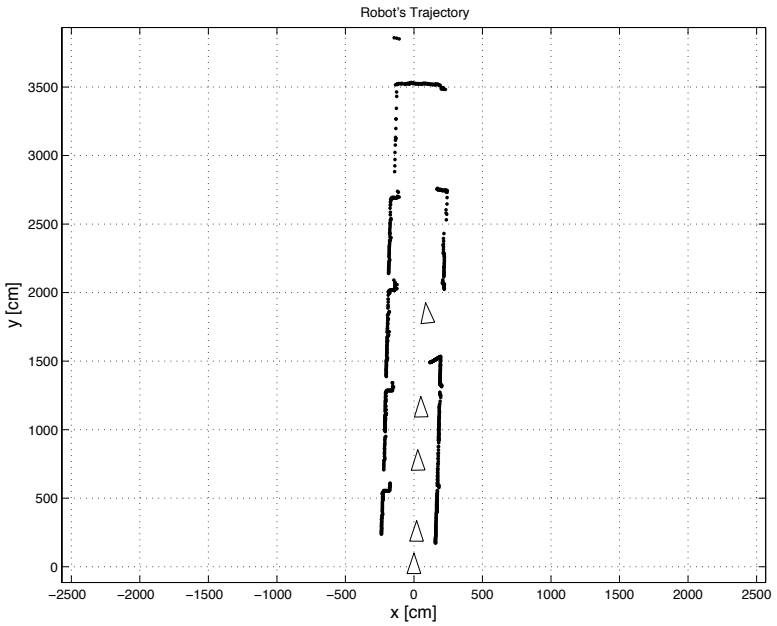



**Figure 4.50:** A robot reconstructing a piece of corridor in the Computer Science Department at the UCLA while localizing itself.

## 4.3 Alternative Loss Functions

With ICPSAC, it is trivial to modify the errror function to include a robust kernel One of the problems with RANSAC is that if the threshold $e_{th}$ for considering the inliers is set too high then the estimate can be very poor. RANSAC, in effect, finds the minimum of a cost function defined as

$$C_1 = \sum_i c_1(e_i^2) \tag{4.15}$$

where $c_1(\cdot)$ is

$$c_1(e^2) = \begin{cases} 0 & e^2 < e_{th}^2 \\ \text{constant} & e^2 \geq e_{th}^2 \end{cases}, \tag{4.16}$$

In other words inliers score nothing and each outlier scores a constant penalty. Thus the higher $e_{th}^2$ is the more solutions with equal values of $C$ tending to poor estimation. Rather than minimizing $C$, a new cost function $C_2$ can be minimized

$$C_2 = \sum_i c_2(e_i^2) \tag{4.17}$$

where the robust error term $c_2(\cdot)$ is

$$c_2(e^2) = \begin{cases} e^2 & e^2 < e_{th}^2 \\ T^2 & e^2 \geq e_{th}^2 \end{cases}, \tag{4.18}$$

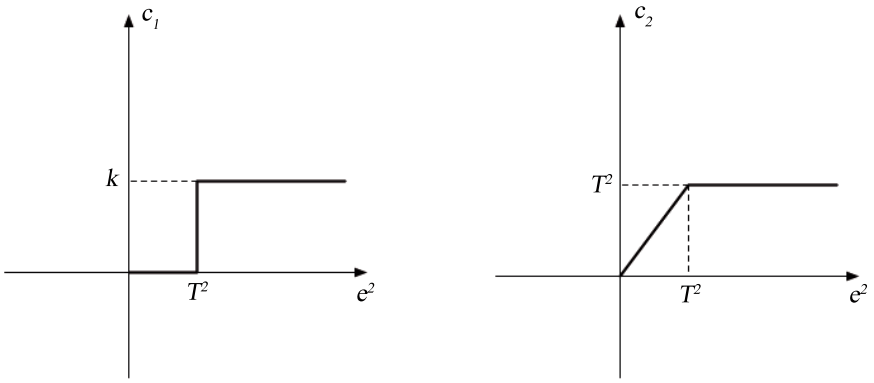
One can see that outliers are still given a fixed penalty but now inliers are scored on how well they fit the data.

For optimizing the efficiency, other loss functions can be used. For instance, the Hampels et al.'s [10] loss function $c_3$ is a variant of the $c_2$, and it is defined as

$$c_3(e^2) = \begin{cases} \frac{1}{2}e^2 & |e| \leq a \\ \frac{1}{2}a(2|e| - a) & a < |e| \leq b \\ \frac{1}{2}a[(|e| - c)^2/(b - c) + (b + c - a)] & b < |e| \leq c \\ \frac{1}{2}a(b + c - a) & c < |e| \end{cases}, \tag{4.19}$$

i.e. it has still a rejection point, beyond which an outlier has no influence. An alternative is the Student's t-distribution [12], whose function is defined as

$$c_4(e) = \frac{1}{2}(1 + f)\log(1 + e^2/f) \tag{4.20}$$




(a) (b)

**Figure 4.51:** Canonical loss functions for determining whether or not a point is compatible with a model.

that does not have a finite rejection point. All these loss functions have the drawback to be nonconvex and therefore may have multiple local minima. A more robust loss function is the Huber's [11] M-estimator, that is defifend as

$$c_5(e^2) = \begin{cases} \frac{1}{2}e^2 & |e| \leq c \\ \frac{1}{2}c(2|e| - c) & |e| > c \end{cases} , \tag{4.21}$$

The last three loss functions are shown in Figure 4.52.

# 4.4 Final Consideration

We conclude this section saying that ICPSAC is a new very robust algorithm that combines the semplicity of ICP with the robustness of RANSAC. Experimental results show that it is very robust regardless to (i) an initial estimate of the relative pose and (ii) noise and (iii) strong outliers presence, having moreover a short convergence time. Above all ICPSAC method simultaneously finds both the rigid transformation and the correspondence between two point sets without making any prior assumption of the transformation model.

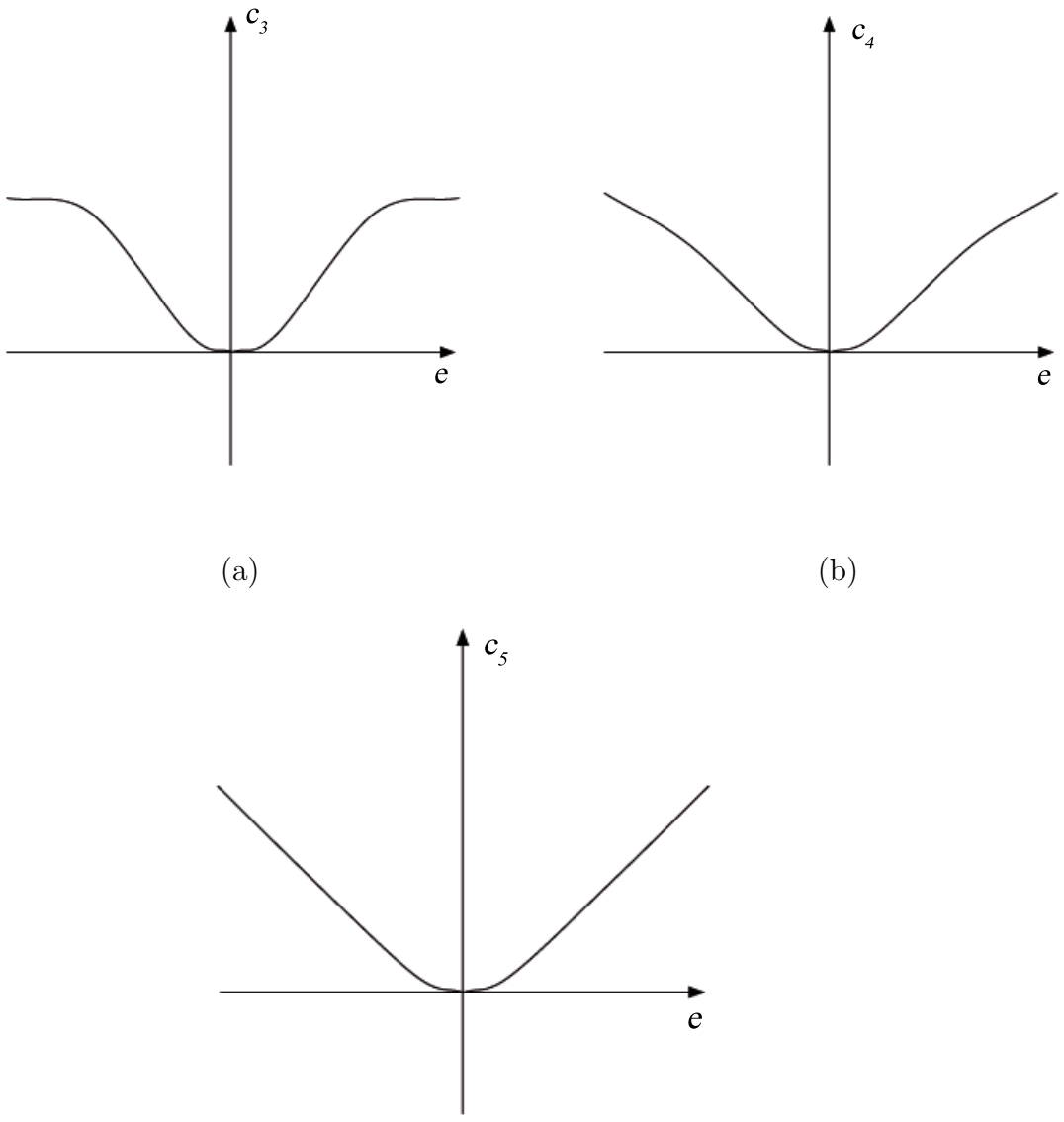



**Figure 4.52:** Alternative loss functions for determining whether or not a point is compatible with a model: (a) *Hampels*, (b) *Student's t-distribution*, (c) *Huber*.

# Chapter 5

# Alternative Algorithms

## 5.1 CPD Algorithm

### 5.1.1 Algorithm description

The Coherent Point Drift (CPD) algorithm is a probabilistic method essentially thought for non-rigid registration but it can be used in the rigid case as well. The registration is treated as a Maximum Likelihood estimation problem with motion coherence constraint over the velocity field such that one point set moves coherently to align with the second set. The idea is, given two point sets, to fit a Gaussian Mixture Model to the first point set, whose Gaussian centroids are initialized from the points in the second set. The process of adapting the Gaussian centroids from their initial position to their final position is viewed as a temporal motion process with the imposed motion coherence constraint, over the velocity field, according to which points close to one another tend to move coherently.

### 5.1.2 Registration results

Simulated performances are shown in the Figure 5.1-5.2 for the noiseless case, whereas the noisy case is shown in the Figure 5.3-5.4. The algorithm is robust to noise and to medium percentages of outliers but it cannot correctly manage medium rotations, e.g. $\theta = -\pi/4$, and high pergentage of outliers. In fact, the failure percentage in such case is 1. Also, CPD registration error is larger than ICPSAC one, it is useless in dynamic environment and computational demanding.

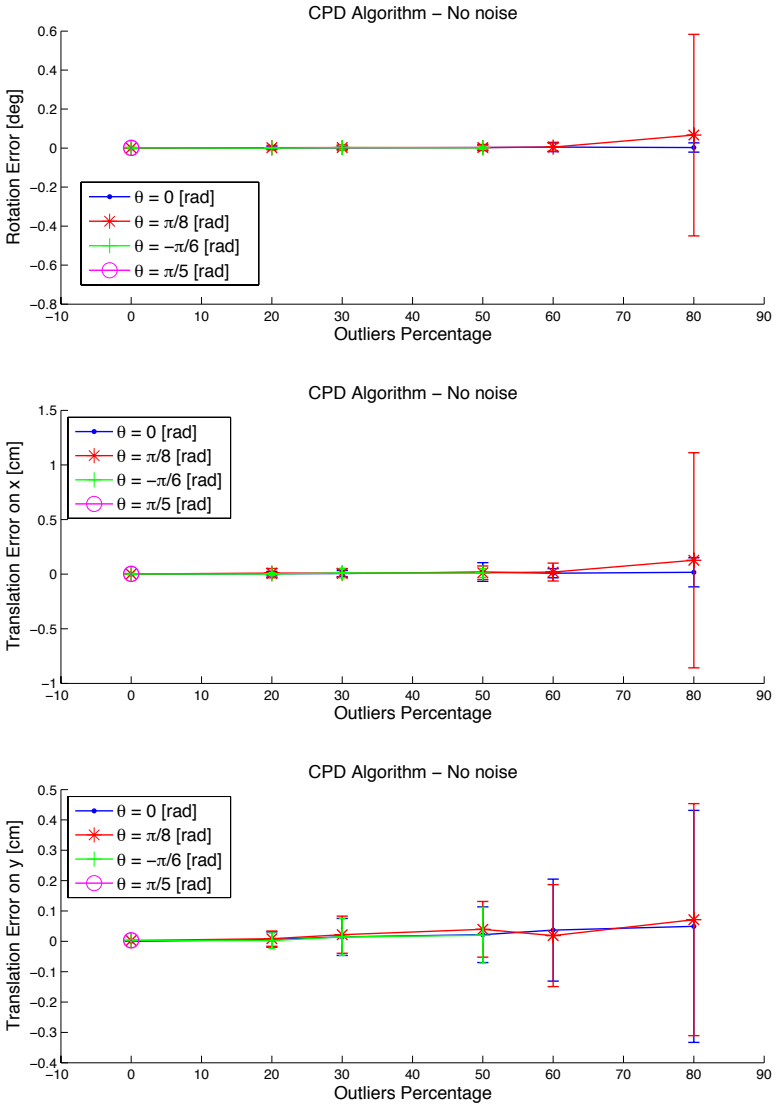



**Figure 5.1:** Average registration error and standard deviation, respectively on $\theta$, $T_x$ and $T_y$, obtained by using the CPD algorithm without noise for different values of the angle rotation $\theta$.

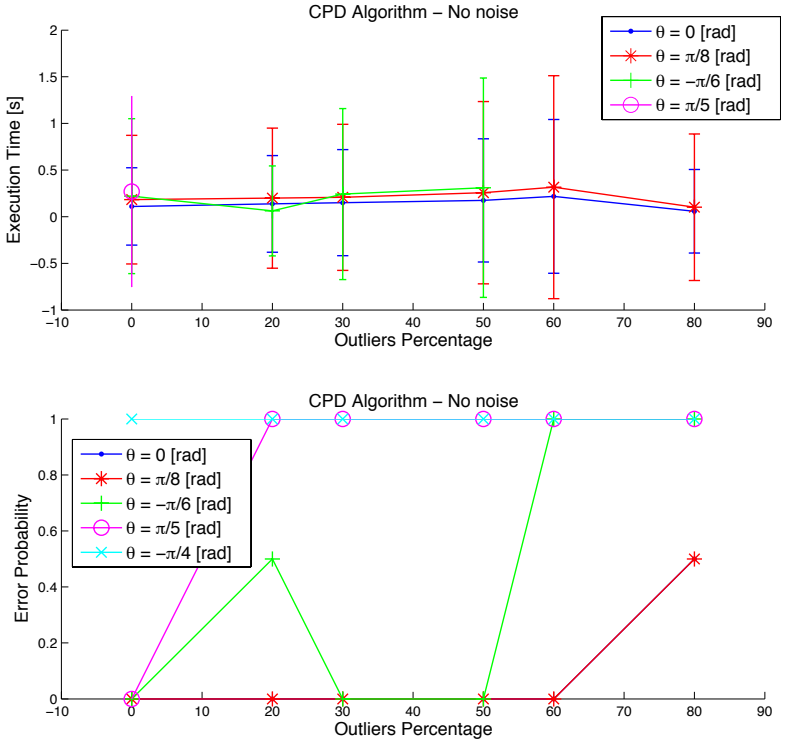



**Figure 5.2:** Execution time and error probability of the CPD algorithm without noise for different values of the angle rotation $\theta$.

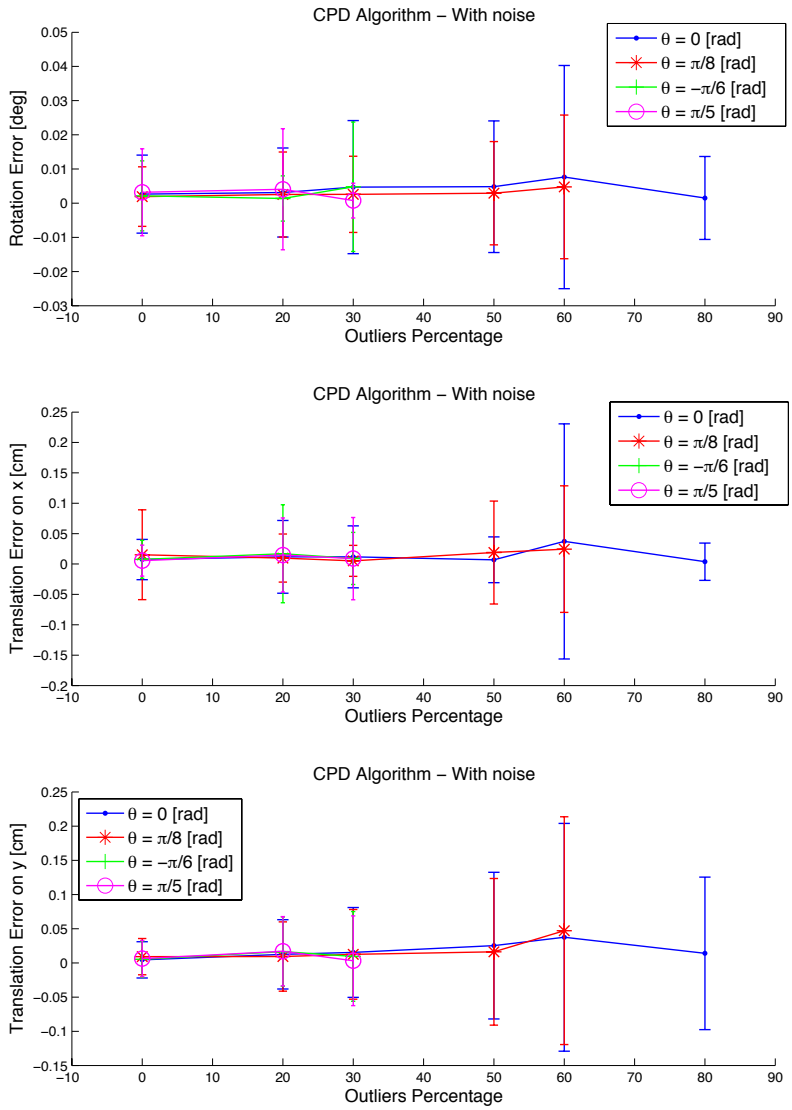



**Figure 5.3:** Average registration error and standard deviation, respectively on $\theta$, $T_x$ and $T_y$, obtained by using the CPD algorithm in presence of noise for different values of the angle rotation $\theta$.

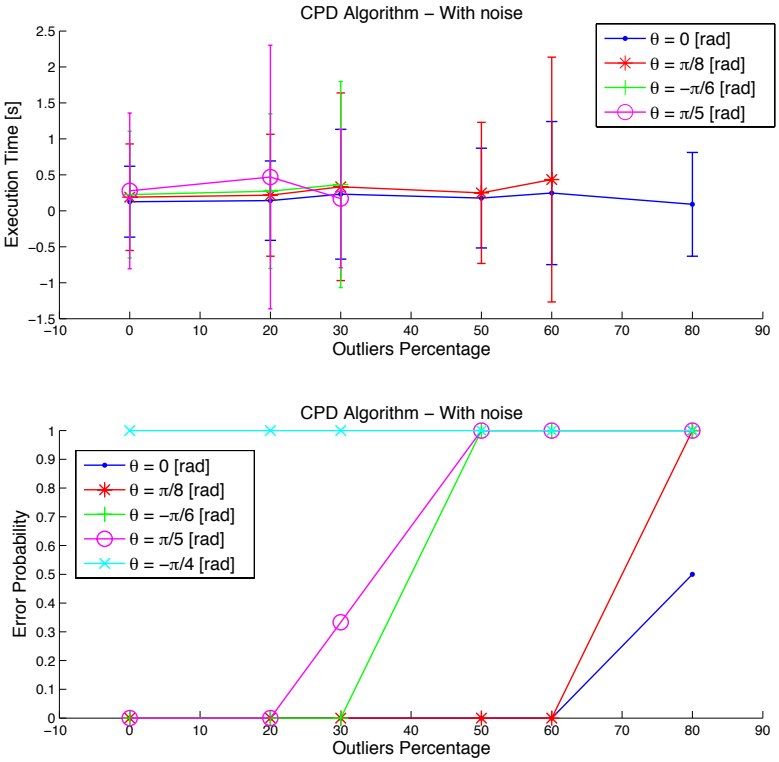



**Figure 5.4:** Execution time and error probability of the CPD algorithm in presence of noise for different values of the angle rotation $\theta$.

## 5.2 Kernel Correlation Algorithm

### 5.2.1 Algorithm description

The Kernel Correlation Algorithm faces the point set registration problem as finding the maximum kernel correlation configuration of the two point sets to be registered. Kernel Correlation (KC) is an affinity measure and on a point set $\mathcal{X}$ it is defined as the total sum of the KC of all the points $\mathbf{q}_k$ in the set,

$$KC(\mathcal{X}) = \sum_i KC(\mathbf{x}^{\mathbf{q}_i}, \mathcal{X}) \,, \tag{5.1}$$

where $KC(\mathbf{x}^{\mathbf{q}_i}, \mathcal{X})$ is defined as

$$KC(\mathbf{x}^{\mathbf{q}_i}, \mathcal{X}) = \sum_{i \neq j} KC(\mathbf{x}^{\mathbf{q}_i}, \mathbf{x}^{\mathbf{q}_j}) \,. \tag{5.2}$$

Finally, KC on two points is defined as

$$KC(\mathbf{x}^{\mathbf{q}_i}, \mathbf{x}^{\mathbf{q}_j}) = \int K(\mathbf{x}, \mathbf{x}^{\mathbf{q}_i}) K(\mathbf{x}, \mathbf{x}^{\mathbf{q}_j}) d\mathbf{x} \,. \tag{5.3}$$

where $K(\mathbf{x}, \mathbf{x}^{\mathbf{q}_i})$ is a kernel function centered at the data point $\mathbf{x}^{\mathbf{q}_i}$. If Gaussian kernel is used, then

$$K_G(\mathbf{x}, \mathbf{x}^{\mathbf{q}_i}) = (\pi\sigma^2)^{-n/2} \exp(-\|\mathbf{x} - \mathbf{x}^{\mathbf{q}_i}\|^2/\sigma^2) \,, \tag{5.4}$$

where $n$ is the vector's dimension and the equation (5.3) can be written as

$$KC_G(\mathbf{x}^{\mathbf{q}_i}, \mathbf{x}^{\mathbf{q}_j}) = (2\pi\sigma^2)^{-n/2} \exp(-\|\mathbf{x}^{\mathbf{q}_i} - \mathbf{x}^{\mathbf{q}_j}\|^2/2\sigma^2) \,. \tag{5.5}$$

At this point, given two finite size point sets, the model set $\mathcal{M}$ and the scene set $\mathcal{S}$, the KC method is defined as finding the parameter vector $\mathbf{a}$ of the transformation $g_{ms}$ which minimizes the following cost function

$$e(\mathcal{S}, \mathcal{M}, \mathbf{a}) = -\sum_{\mathbf{x}_s \in \mathcal{S}} \sum_{\mathbf{x}_m \in \mathcal{M}} KC(\mathbf{x}_m, g_{ms}(\mathbf{x}_s, \mathbf{a})) \,, \tag{5.6}$$

where each transformed scene point $\mathbf{x}_s$ is interacting with all the scene points.

It can be easily shown that

$$KC(\mathcal{M} \cup g_{ms}(\mathcal{S}, \mathbf{a})) = KC(\mathcal{M}) + KC(g_{ms}(\mathcal{S}, \mathbf{a})) - 2e(\mathcal{S}, \mathcal{M}, \mathbf{a}) \,. \tag{5.7}$$

where $KC(\mathcal{M})$ is independent from $\mathbf{a}$ and under rigid transformation, $KC(g_{ms}(\mathcal{S}, \mathbf{a}))$ is constant. Also, it can be shown that the equation (5.6) can be rewritten as

$$e(\mathcal{S}, \mathcal{M}, \mathbf{a}) = -N^2 \int_{\mathbf{x}} P_{\mathcal{M}} P_{\mathcal{S}} d\mathbf{x} \,, \tag{5.8}$$

where

$$P_{\mathcal{M}}(\mathbf{x}) = \sum_{\mathbf{x}_m \in \mathcal{M}} K(\mathbf{x}, \mathbf{x}_m)/N\,, \quad P_{\mathcal{S}}(\mathbf{x}, \mathbf{a}) = \sum_{\mathbf{x}_s \in \mathcal{S}} K(\mathbf{x}, g_{ms}(\mathbf{x}_s, \mathbf{a}))/N\,, \quad (5.9)$$

are the kernel density estimates.

### 5.2.2 Registration results

As for CPD algorithm case, simulated performances are given. The noiseless case is shown in the Figure 5.5-5.6, whereas the noisy case is shown in the Figure 5.7-5.8. KC cannot manage outliers as well as CPD and moreover its registration error is much larger than ICPSAC one. Also is computational demanding and useless in dynamic environment.

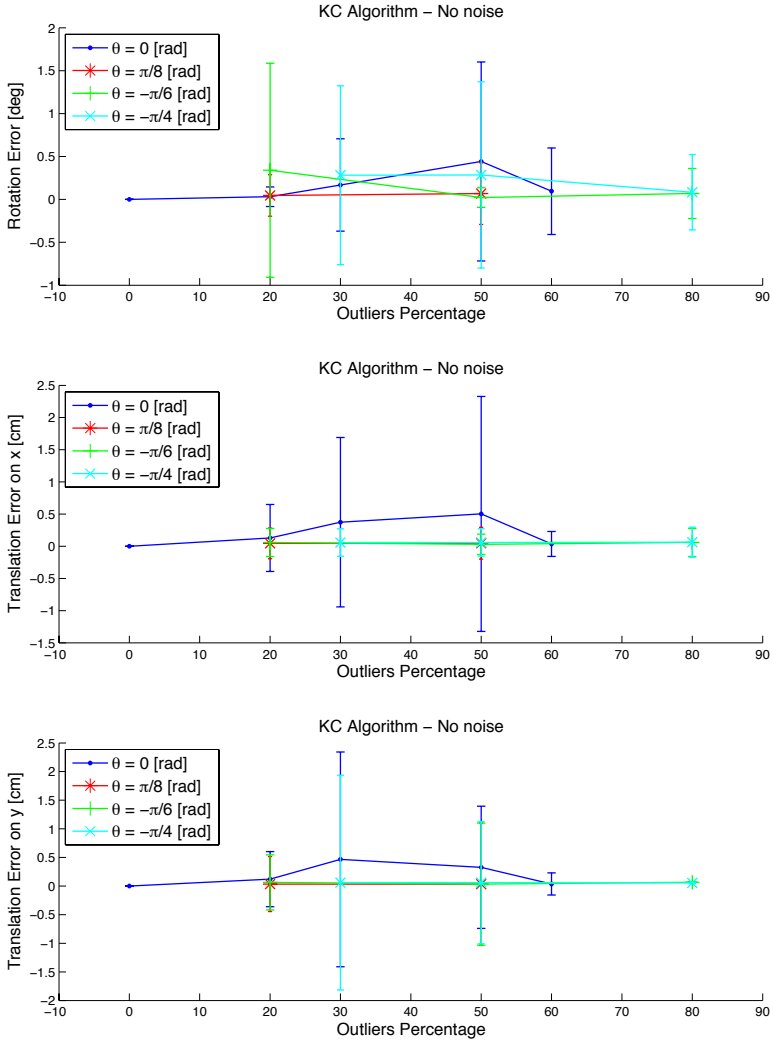



**Figure 5.5:** Average registration error and standard deviation, respectively on $\theta$, $T_x$ and $T_y$, obtained by using the KC algorithm without noise for different values of the angle rotation $\theta$.

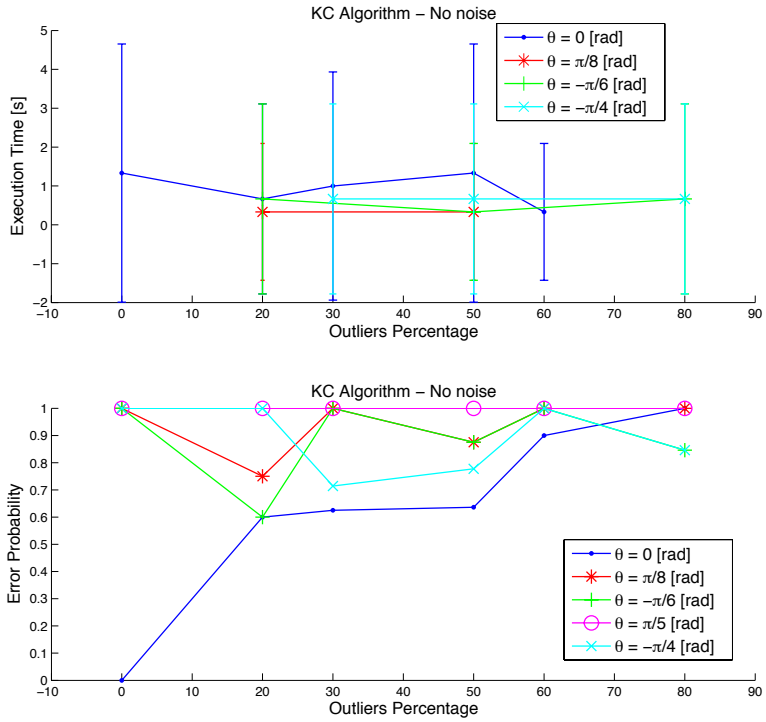



**Figure 5.6:** Execution time and error probability of the KC algorithm without noise for different values of the angle rotation $\theta$.

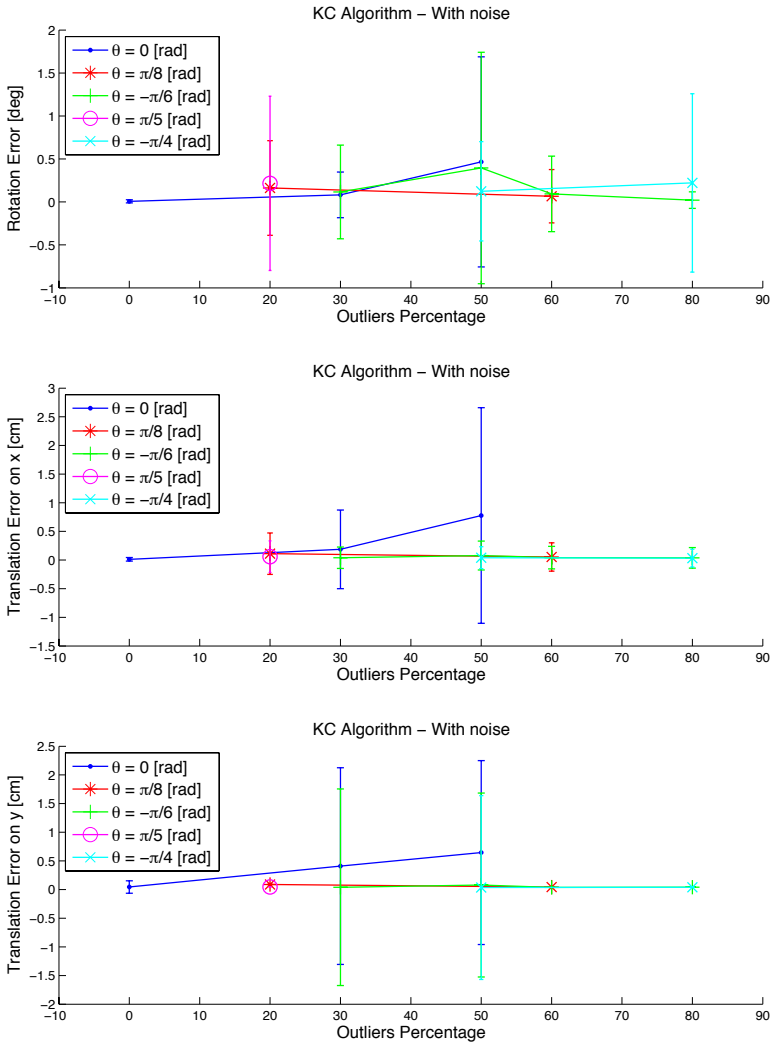



**Figure 5.7:** Average registration error and standard deviation, respectively on $\theta$, $T_x$ and $T_y$, obtained by using the KC algorithm in presence of noise for different values of the angle rotation $\theta$.

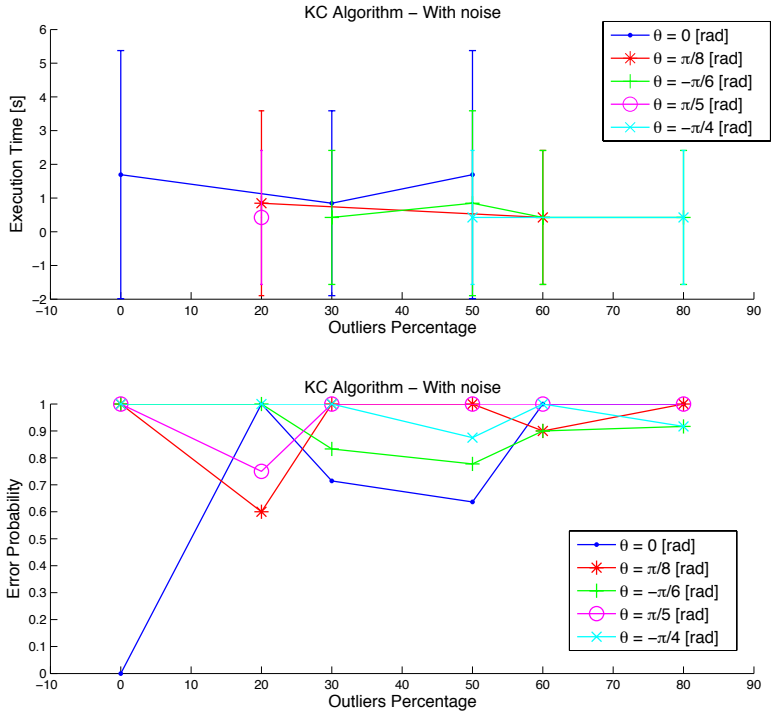



**Figure 5.8:** Execution time and error probability of the KC algorithm in presence of noise for different values of the angle rotation $\theta$.

# Chapter 6

# Conclusion

Point set registration is a problem of pivotal importance that continues to attract considerable interest. In this work we present a novel algorithm for point set registration which unifies some previous works in the field, that exhibits better tolerance to outliers and is more computationally efficient than competiting methods.

Most of the existing techniques for solving the partially-overlapping registration problem have the following limitations:

- they cannot ensure a correct solution even for the noiseless case;
- they require a good initial estimate of the rigid transformation between the two data sets.

In this thesis, we have proposed the ICPSAC approach, which has none of the above limitations. It can be used for the featureless case while requiring no initial estimates. Furthermore, our method is faster than most of the existing methods. Also, it uses the rigidity constraint among the pre-selected points to restrict the possible associations. Experiments on both syntethic and real data have demonstrated the effectiveness of our method and that it is efficient and reliable for registering lowly overlapping range images.

# Appendix A

# Corresponding Point Set Registration

The closed form procedures for yielding the least squares rotation and translation is reviewed. The quaternion-based algorithm is preferred in three dimensions, while in two dimensions we use a simple algorithm that finds the oriented angle between two vectors.

## A.1 Quaternion-based Algorithm

The unit quaternion is a vector $\mathbf{q}_R = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T \in \mathbb{R}^4$, where $q_0 \geq 0$, and $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. The rotation matrix $R \in SO(3)$ generated by a unit rotation quaternion is

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix}. \tag{A.1}$$

Let $\mathbf{q}_T = \begin{bmatrix} q_4 & q_5 & q_6 \end{bmatrix}^T$ be a translation vector. Now, the complete registration state vector is $\mathbf{q} = \begin{bmatrix} \mathbf{q}_R & \mathbf{q}_T \end{bmatrix}^T$.

Let $\mathcal{S} = \{\mathbf{s}_i\}_{i=1}^{N_s}$ be a measured data point set to be aligned with a model point set $\mathcal{M} = \{\mathbf{m}_i\}_{i=1}^{N_m}$ where $N_s = N_m = N$ and where each point $\mathbf{s}_i$ corresponds to the point $\mathbf{m}_i$ with the same index. The function to be minimized is

$$f(\mathbf{q}) = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{x}^{\mathbf{m}_i} - R(\mathbf{q}_R)\mathbf{x}^{\mathbf{s}_i} - \mathbf{q}_T \right\|^2. \tag{A.2}$$

The center of mass $\mu_s$ and $\mu_m$ are given by

$$\mu_s = \frac{1}{N_s}\sum_{i=1}^{N_s}\mathbf{x}^{\mathbf{s}_i}\,, \qquad \mu_m = \frac{1}{N_m}\sum_{i=1}^{N_m}\mathbf{x}^{\mathbf{m}_i}\,. \tag{A.3}$$

The cross-covariance matrix $\Sigma$ of the sets $\mathcal{S}$ and $\mathcal{M}$ is then given by

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}[(\mathbf{x}^{\mathbf{s}_i}-\mu_s)(\mathbf{x}^{\mathbf{m}_i}-\mu_m)^T] = \frac{1}{N}\sum_{i=1}^{N}[(\mathbf{x}^{\mathbf{s}_i})(\mathbf{x}^{\mathbf{m}_i})^T] - \mu_s\mu_m^T\,. \tag{A.4}$$

The vector $\Delta = [A_{23}\ A_{31}\ A_{12}]^T$ where $A_{ij} = (\Sigma - \Sigma^T)_{ij}$, is used to form the symmetric matrix $Q(\Sigma)$

$$Q(\Sigma) = \begin{bmatrix} tr(\Sigma) & \Delta^T \\ \Delta & \Sigma^T - tr(\Sigma)I_3 \end{bmatrix}\,, \tag{A.5}$$

where $I_3$ is the $3\times 3$ identity matrix. The unit eigenvector $\mathbf{q}_R$ corresponding to the maximum eigenvalue of the matrix $Q(\Sigma)$ is selected as the optimal rotation. The optimal translation vector is then given by

$$\mathbf{q}_T = \mu_m - R(\mathbf{q}_R)\mu_s\,. \tag{A.6}$$

## A.2 Vector-based Algorithm

In the 2-D case, the transformation between two point sets is found by using a simpler procedure that needs only two couple of corresponding points. Given two points $\mathbf{m}_1$ and $\mathbf{m}_2$ belonging to the model set and the corresponding scene set points $\mathbf{s}_1$ and $\mathbf{s}_2$, we want to find the transformation that aling the two couple of points.

First of all, we found the vectors $\vec{v}_1$ and $\vec{v}_2$ defined by the couple of points, that is

$$\vec{v}_1 = \mathbf{x}^{\mathbf{m}_1} - \mathbf{x}^{\mathbf{m}_2}\,, \qquad \vec{v}_2 = \mathbf{x}^{\mathbf{s}_1} - \mathbf{x}^{\mathbf{s}_2}\,, \tag{A.7}$$

then the unsigned angle $\bar{\theta}$, i.e. the angle between the two vectors, can be found using the inner product relation

$$\bar{\theta} = \arccos\Big(\frac{\langle\vec{v}_1,\vec{v}_2\rangle}{\|\vec{v}_1\|\|\vec{v}_2\|}\Big)\,. \tag{A.8}$$

Now, for finding the right sign to give to $\bar{\theta}$ for finding $\theta_{ms}$, we check if

$$\text{sign}(\langle\vec{v}_1,\vec{v}_2\rangle) \neq \text{sign}(\langle\vec{v}_1,\vec{v}_3\rangle)\,, \tag{A.9}$$

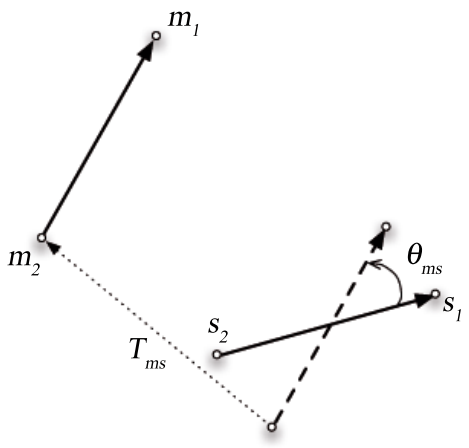



**Figure A.1:** A rigid body transformation between a scene frame $< S >$ and a model frame $< M >$.

where $\vec{v}_3$ is $\vec{v}_2$ rotated of $\pi/2$. In such case the signed angle is $\theta_{ms} = -\bar{\theta}$. Otherwise $\theta_{ms}$ is simply equal to $\bar{\theta}$.

At this point, the translation can be found and it is equal to

$$T_{ms} = \mathbf{x}_m^{\mathbf{m}_1} - R_{ms}\mathbf{x}_s^{\mathbf{s}_1}\,, \tag{A.10}$$

where $R_{ms} \in SO(2)$ is the rotation matrix obtained by $\theta_{ms}$.

# Bibliography


[1] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration", *ECCV2004(3)*, pp.558-569.

[2] B. Jian and B.C. Vemuri, "A Robust Algorithm for Point Set Registration Using Mixture of Gaussians", *ICCV*, 2005.

[3] P. Besl and N. McKay, "A Method for Registration of 3-D Shapes", *Trans. PAMI*, Vol. 14, No. 2, 1992.

[4] A. Fitzgibbon, "Robust Registration of 2D and 3D Point Sets", *British Machine Vision Conference*, 2001.

[5] D. Chetverikov, D. Svirko, D. Stepanov and P. Krsek, "The Trimmed Iterative Closest Point Algorithm", *Proc. ICPR*, 2002.

[6] M. A. Fischler, R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *CACM*, Vol. 24, No. 6, 1981.

[7] M. Greenspan, M. Yurick, "Approximate K-D Tree Search for Efficient ICP", *Proc. IEEE 3DIM*, 2003.

[8] K. Pulli, "Multiview Registration for Large Data Sets", *3DIM*, 1999.

[9] S. Thrun, A. Nuchter, H. Surmann, K. Lingemann, J. Hertzberg, "6D SLAM with an Application in Autonomous Mine Mapping", *Proc. IEEE (ICRA)*, 2004.

[10] F.R. Hampel, P.J. Rousseeuw and E. Ronchetti, "The Change-of-Variance Curve and Optimal Redescending M-estimators", *J. Am. Statistical Assoc.*, Vol. 76, 1981.

[11] P.J. Huber, "Robust Statistics", *New York: John Wiley & Sons*, 1981.

[12] K.L. Boyer, M.J. Mirza and G. Ganguly, "The Robust Sequential Estimator: A General Approach and Its Application to Surface Organization in Range Data", *IEEE Trans. PAMI*, Vol. 17, 1995.

[13] J. Matas and O. Chum, "Randomized RANSAC with Sequential Probability Ratio Test", *ICCV*, 2005.

[14] S. Rusinkiewicz and M. Levoy, "Efficient Variants of the ICP Algorithm", *Proc. IEEE (3-D Digital Imaging and Modeling)*, 2001.

[15] C.Chen, Y. Hung and J. Cheng, "RANSAC-based DARCES: A New Approach to Fast Automatic Registration of Partially Overlapping Range Images", *Proc. of ICCV*, 1998.

[16] B. Tordoff and D.W. Murray, "Guided sampling and consensus for motion estimation", *Proc. European Conference on Computer Vision*, 2002.

[17] P.H.S. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry", *Computer Vision and Image Understanding*, 1996.

[18] D. Nistér, "Preemptive RANSAC for Live Structure and Motion Estimation", *Proc. of ICCV*, 2003.

[19] B.K.P. Horn, "Closed-form solution of absolute orientation using unit quaternions", *Journal of the Optical Society of America*, 1987.

[20] A. Myronenko, X. Song and M.Á. Carreira-Perpinán, "Non-rigid point set registration: Coherent Point Drift", *NIPS Conference*, 2006.