Using Augmented Measurements to Improve the Convergence of ICP

Jacopo Serafin, Giorgio Grisetti

Dept. of Computer, Control and Management Engineering,
Sapienza University of Rome, Via Ariosto 25, I-00185, Rome, Italy
{serafin,grisetti}@dis.uniroma1.it

Abstract. Point cloud registration is an essential part for many robotics applications and this problem is usually addressed using some of the existing variants of the Iterative Closest Point (ICP) algorithm. In this paper we propose a novel variant of the ICP objective function which is minimized while searching for the registration. We show how this new function, which relies not only on the point distance, but also on the difference between surface normals or surface tangents, improves the registration process. Experiments are performed on synthetic data and real standard benchmark datasets, showing that our approach outperforms other state of the art techniques in terms of convergence speed and robustness.

Keywords: Point Cloud Registration · ICP · Surface Normals

1 Introduction

Registering two point clouds is a building block of many robot applications such as simultaneous localization and mapping (SLAM), object recognition and detection, augmented reality and many others. This problem is commonly solved by variants of the Iterative Closest Point (ICP) algorithm proposed by Besl and McKay [1]. ICP tries to find a transformation that minimizes the distance of a set of corresponding points in the two clouds. At each iteration ICP refines the estimate of the transformation by alternating a search and an optimization routine. Given the current transform, the search looks for corresponding points in the two clouds. The optimization computes the transformation that results in the minimum distance between the corresponding points found by the search step. ICP is a very successful scheme and several variants of increasing performances have been proposed. If the correspondences are free from outliers and the measurements are affected by low noise, the transformation can be found directly by applying the Horn formula [3].

The whole concept at the base of ICP is that, at each iteration, an improved transformation with respect to the previous one is found. Such a transformation represents the new initial guess for the heuristic used to find the correspondences and allows to determine better associations at the next iteration. Accordingly, researchers focused on seeking for heuristics that provide "good" correspondences.

The original idea of picking up the closest points [1] has been progressively refined to consider features, curvature and other characteristics of the points. Pomerlau *et al.* [5] provided an excellent overview on these different variants.

ICP and its variants require multiple iterations because it does not exist an heuristic that provides the exact correspondences. Since the optimization requires linear time in the number of correspondences, the bottleneck of the computation is represented by the heuristic that has to compute them.

The main drawback of ICP in its original formulation is the assumption that the points in the two surfaces are exactly the same. This is clearly not true as the point clouds are obtained by sampling a set of points from the surface observed by the sensor. If the observer position changes, the chances that two points in the clouds are the same is very low. This is particularly evident at low sampling resolutions. Aware of this aspect, Magnusson *et al.* [4] proposed to approximate the surface with a set of Gaussians capturing the local statistics of the surface in the neighborhood of a point. In that representation, called the Normal Distrubution Transform (NDT), the correspondence search uses the Mahalanobis distance instead of the Euclidean one and the optimization tries to minimize it.

Similarly, Segal *et al* [6] proposed a refined version of ICP called *Generalized ICP* (GICP). The core idea behind this algorithm is to account for the shape of the surface which surrounds a point by approximating it with a planar patch. In the optimization, two corresponding patches are aligned onto each other, neglecting the error along their tangent direction. This can be straightforwardly implemented by minimizing the Mahalanobis distance of corresponding points, where the covariance matrix of a measurement is forced to have the shape of a disk aligned with the sampled surface. Thanks to the better rejection of false correspondences based on the surface normal cue and the more realistic objective function, NDT and GICP exhibit a substantially more stable convergence behavior.

In fact, within ICP and its variants, the optimization and the correspondence search are not independent. If the optimization is robust to outliers and exhibits a smooth behavior, the chances that it finds a better solution at the subsequent step increases. In this way an improvement is obtained at each iteration until a good solution is found. Despite NDT and GICP, the authors are unaware of other methods that improve the objective function.

Since point clouds are the effects of sampling a surface, the local characteristics of this surface play a role in the optimization. From this point of view, the objective function has to express some distance between *surface samples*, and the optimization algorithm has to determine the optimal alignment between these two set of samples. A surface sample, however, is not fully described just by 3D points, but it requires additional cues like the surface normal, the curvature and, potentially, the direction of the edge. Both NDT and GICP minimize a distance between corresponding points, while they neglect additional cues that can indeed play a role in determining the transformation and in rejecting outliers.

In this paper we propose a novel variant of the objective function which is optimized while searching for the transformation. This function depends not only on the relative point distance, but also on the difference between surface normals or tangents in case the point lies on an edge. We provide an iterative form for the optimization routine and we show through experiments performed on synthetic data and standard benchmark datasets that our approach outperforms other state of the art techniques, both in terms of convergence speed and robustness.

2 ICP

The problem of registering two point clouds consists in finding the rotation and the translation that maximizes the overlap between the two clouds. More formally, let $\mathcal{P}^{\mathrm{r}} = \{\mathbf{p}^{\mathrm{r}}_{1:N^{\mathrm{r}}}\}$ and $\mathcal{P}^{\mathrm{c}} = \{\mathbf{p}^{\mathrm{c}}_{1:N^{\mathrm{c}}}\}$ be the two set of points, we want to find the transformation $\mathbf{T}^*$ that minimizes the distance between corresponding points in the two scenes:

$$\mathbf{T}^* = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_{\mathcal{C}} \overbrace{\left(\mathbf{p}^{\mathrm{c}}_i - \mathbf{T} \oplus \mathbf{p}^{\mathrm{r}}_j\right)^T \mathbf{\Omega}_{ij} \underbrace{\left(\mathbf{p}^{\mathrm{c}}_i - \mathbf{T} \oplus \mathbf{p}^{\mathrm{r}}_j\right)}_{\mathbf{e}_{ij}(\mathbf{t})}}^{\chi^2_{ij}} \ . \qquad (1)$$

In Eq. 1 the symbols have the following meaning:

- $\mathbf{T}$ is the transform that is updated at each step i of the iterative algorithm with the one found at iteration $i-1$;
- $\mathbf{\Omega}_{ij}$ is an information matrix that takes into account the noise properties of the sensor or of the surface;
- $\mathcal{C} = \{\langle i,j\rangle_{1:M}\}$ is a set of correspondences between points in the two clouds. $\langle i,j\rangle \in \mathcal{C}$ means that the point $\mathbf{p}^{\mathrm{r}}_j$ in the cloud $\mathcal{P}^{\mathrm{r}}$ corresponds to the point $\mathbf{p}^{\mathrm{c}}_i$ in the cloud $\mathcal{P}^{\mathrm{c}}$;
- $\mathbf{e}_{ij}(\mathbf{t})$ is the error function that computes the distance between the point $\mathbf{p}^{\mathrm{c}}_i$ and the corresponding point $\mathbf{p}^{\mathrm{r}}_j$ in the other cloud after applying the transformation $\mathbf{T}$;
- χ^2_{ij} is the $\mathbf{\Omega}_{ij}$-norm of the error $\mathbf{e}_{ij}(\mathbf{t})$;
- $\oplus$ is an operator that applies the transformation $\mathbf{T}$ to a point $\mathbf{p}$. If we use the homogeneous notation for transformations and points, $\oplus$ reduces to the matrix-vector product.

In general, the correspondences between two point clouds are not known. However, in presence of a good approximation for the initial transform, they can be “guessed” through some heuristic like nearest neighbour. In its most general formulation, ICP iteratively refines an initial transform $\mathbf{T}$ by searching for correspondences and finding the solution of Eq. 1. Such a new transformation is then used in order to compute the new correspondences. Eq. 1 describes the objective function used in the optimization of ICP, NDT and GICP. In the case of ICP, $\mathbf{\Omega}_{ij}$ is a diagonal matrix potentially scaled with a weight representing the confidence about the correctness of a correspondence. NDT computes the covariances $\mathbf{\Sigma}_i$ directly from the point cloud and it measures the distances by using the mean of the Gaussians rather than the points as shown in Eq. 2.

$$\mathbf{T}^* = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_{\mathcal{C}} \left(\mu_i^{\mathrm{c}} - \mathbf{T} \oplus \mathbf{p}_j^{\mathrm{r}}\right)^T \mathbf{\Sigma}_i^{-1} \left(\mu_i^{\mathrm{c}} - \mathbf{T} \oplus \mathbf{p}_j^{\mathrm{r}}\right) \ . \qquad (2)$$

In GICP, $\mathbf{\Omega}_{ij} = \mathbf{\Sigma}_i^{-1}$ depends only on the i^{th} point $\mathbf{p}_i^{\mathrm{c}}$ and its neighborhood. The covariance $\mathbf{\Sigma}_i$ is enforced to have a disk shape and to lie on the surface from where $\mathbf{p}_i^{\mathrm{c}}$ was sampled. In all cases, the difference $\mathbf{p}_i^{\mathrm{c}} - \mathbf{T} \oplus \mathbf{p}_j^{\mathrm{r}}$ is a 3D vector that measures the offset between two 3D points and the domain of the error function is $\Re^3$.

Since an increase in the dimensionality of the points makes the whole system more observable, less correspondences are required for the optimization process. By characterizing each point with other quantities to which a transform can be applied, we can achieve such an increase in the dimensionality. We propose, for this reason, the use of normals for quasi-planar regions and/or tangents for regions of high curvature.

3 Extended ICP

In this section we describe the extension of the model of ICP in order to consider also normals and tangents of the surface. We first illustrate the general concept and, subsequently, we focus on the case in which a local surface has either a normal, a tangent or none of the two. We conclude the section by sketching an algorithm to carry on the optimization.

3.1 Extending the Measurements

Let $\mathbf{n}_i$ be the normal of a point $\mathbf{p}_i$ belonging to a certain surface, and $\boldsymbol{\tau}_i$ its tangent if the point is part of an edge, we can then extend Eq. 1 as follows:

$$\begin{aligned} \mathbf{T}^* = \underset{\mathbf{T}}{\operatorname{argmin}} \quad & \sum_{\mathcal{C}} \left(\mathbf{p}_i^{\mathrm{c}} - \mathbf{T} \oplus \mathbf{p}_j^{\mathrm{r}}\right)^T \mathbf{\Omega}_{ij}^{\mathrm{p}} \left(\mathbf{p}_i^{\mathrm{c}} - \mathbf{T} \oplus \mathbf{p}_j^{\mathrm{r}}\right) \\ + \quad & \sum_{\mathcal{C}} \left(\mathbf{n}_i^{\mathrm{c}} - \mathbf{T} \oplus \mathbf{n}_j^{\mathrm{r}}\right)^T \mathbf{\Omega}_{ij}^{\mathrm{n}} \left(\mathbf{n}_i^{\mathrm{c}} - \mathbf{T} \oplus \mathbf{n}_j^{\mathrm{r}}\right) \\ + \quad & \sum_{\mathcal{C}} \left(\boldsymbol{\tau}_i^{\mathrm{c}} - \mathbf{T} \oplus \boldsymbol{\tau}_j^{\mathrm{r}}\right)^T \mathbf{\Omega}_{ij}^{\tau} \left(\boldsymbol{\tau}_i^{\mathrm{c}} - \mathbf{T} \oplus \boldsymbol{\tau}_j^{\mathrm{r}}\right) \ . \end{aligned} \qquad (3)$$

Here $\mathbf{n}_i^{\mathrm{c}}$, $\mathbf{n}_j^{\mathrm{r}}$ and $\mathbf{\Omega}_{ij}^{\mathrm{n}}$ represent respectively the normal of the point $\mathbf{p}_i^{\mathrm{c}}$ and $\mathbf{p}_i^{\mathrm{r}}$, and the information matrix of the correspondence among the two normals. Similarly, $\boldsymbol{\tau}_i^{\mathrm{c}}$, $\boldsymbol{\tau}_j^{\mathrm{r}}$ and $\mathbf{\Omega}_{ij}^{\tau}$ are the tangents and the information matrix of the correspondence among the two tangents. We recall that, if $\mathbf{T}$ is a transformation described by a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$, the $\oplus$ operator has different definitions depending on its arguments:

$$\mathbf{T} \oplus \mathbf{x} = \begin{cases} \mathbf{R} \cdot \mathbf{x} + \mathbf{t} & \text{if } \mathbf{x} \text{ is a point} \\ \mathbf{R} \cdot \mathbf{x} & \text{if } \mathbf{x} \text{ is a tangent or a normal} \end{cases} \qquad (4)$$

A Mahalanobis distance between two point clouds can be measured by considering also the *distances of corresponding normals* and *corresponding tangents* after applying the transformation $\mathbf{T}$, as shown in Eq. 3.

By defining an extended point $\bar{\mathbf{p}}$ as a vector consisting of a point $\mathbf{p}$, its normal $\mathbf{n}$ and its tangent $\boldsymbol{\tau}$, we have a straightforward modification of the $\oplus$ operator as

$$\bar{\mathbf{p}} = \begin{pmatrix} \mathbf{p}, \mathbf{n}, \boldsymbol{\tau} \end{pmatrix}^T \qquad \mathbf{T} \oplus \bar{\mathbf{p}} = \begin{pmatrix} \mathbf{R}\mathbf{p} + \mathbf{t}, \mathbf{R}\mathbf{n}, \mathbf{R}\boldsymbol{\tau} \end{pmatrix}^T \ . \tag{5}$$

Eq. 3 can be, then, compactly rewritten in terms of extended points as

$$\mathbf{T}^* = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_{\mathcal{C}} \left(\bar{\mathbf{p}}_i^{\mathrm{c}} - \mathbf{T} \oplus \bar{\mathbf{p}}_j^{\mathrm{r}} \right)^T \bar{\boldsymbol{\Omega}}_{ij} \left(\bar{\mathbf{p}}_i^{\mathrm{c}} - \mathbf{T} \oplus \bar{\mathbf{p}}_j^{\mathrm{r}} \right) \ , \tag{6}$$

where $\bar{\boldsymbol{\Omega}}_{ij} = \operatorname{diag}(\boldsymbol{\Omega}_{ij}^{\mathrm{p}}, \boldsymbol{\Omega}_{ij}^{\mathrm{n}}, \boldsymbol{\Omega}_{ij}^{\tau})$ summarizes the contribution of $\boldsymbol{\Omega}_{ij}^{\mathrm{p}}$, $\boldsymbol{\Omega}_{ij}^{\mathrm{n}}$ and $\boldsymbol{\Omega}_{ij}^{\tau}$. The function $\operatorname{diag}(\mathbf{a}_1, \ldots, \mathbf{a}_k)$ stands for a diagonal matrix whose entries, starting from the upper left corner, are $\mathbf{a}_1, \ldots, \mathbf{a}_k$. If the point is not sampled from a locally planar surface nor from an edge, a reasonable distance metric is the Euclidean distance. For these points, we fall back to the ICP case, which is enclosed in Eq. 6 by setting the information matrices of the tangent and the normal to the null matrix: $\boldsymbol{\Omega}_{ij}^{\mathrm{n}} = \boldsymbol{\Omega}_{ij}^{\tau} = 0$.

When measuring the distance between two planar patches, it is reasonable to neglect displacements along the tangent direction of the plane, while errors along the normal direction should be more severely penalized. Additionally, the normals of the two planes should be as close as possible. However, *this constraint cannot be enforced when using only 3D points*. To obtain this behavior from the error function, we can impose ${\boldsymbol{\Omega}_{ij}^{\mathrm{p}}}^{-1}$ to be a disc lying on the surface around $\mathbf{p}_i^{\mathrm{c}}$, as done in [6]. Since the tangent is not defined in a planar patch, we set $\boldsymbol{\Omega}_{ij}^{\tau} = 0$. Additionally, we set the covariance matrix ${\boldsymbol{\Omega}_{ij}^{\mathrm{n}}}^{-1}$ of the normal to have a shape which is elongated in the normal direction. In this way the error between the normals introduces a strong momentum that "forces" them to have the same direction.

Conversely, when measuring the distance between two edges, it is reasonable to slide them onto each other along the tangent direction. This behavior can be obtained by enforcing ${\boldsymbol{\Omega}_{ij}^{\mathrm{p}}}^{-1}$ to have a prolonged shape and to lie along the tangent direction. The tangents $\boldsymbol{\tau}$, instead, can be used to penalize two edges not lying on the same direction by setting $\boldsymbol{\Omega}_{ij}^{\tau}$ to have a shape which is elongated in the direction of $\boldsymbol{\tau}$. Since an edge has no normal, $\boldsymbol{\Omega}_{ij}^{\mathrm{n}}$ has to be set to 0.

The reader might notice that tangents and normals are mutually exclusive. Since the contributions of the tangent and the normal components to the χ_{ij}^2 have the same matrix dimensions, we can further simplify the extended point $\bar{\mathbf{p}}$ by partitioning it into an affine part $\mathbf{p}$, and in a linear part $\mathbf{l}$. The former is subject to translations and rotations, the latter only to the rotation. In this way it is possible to reduce the dimension of the error function and to speed up the calculation without loss of generality. We therefore define a compact form for an extended point $\tilde{\mathbf{p}}$ as

$$\tilde{\mathbf{p}} = \begin{pmatrix} \mathbf{p}, \mathbf{l} \end{pmatrix}^T \qquad \mathbf{T} \oplus \tilde{\mathbf{p}} = \begin{pmatrix} \mathbf{R}\mathbf{p} + \mathbf{t}, \mathbf{R}\mathbf{l} \end{pmatrix}^T \ . \tag{7}$$

Table 1: This table summarizes the components of the information matrix used in our algorithm, depending on the type of the structure around a point. $\mathbf{R}_{n_i}$ and $\mathbf{R}_{\tau_i}$ are two rotation matrices that bring the y axis respectively along the direction of the normal $\mathbf{n}_i$, or the tangent $\boldsymbol{\tau}_i$. ϵ is a small value (10^{-3} in our experiments).

Case	$\boldsymbol{\Omega}^{\mathrm{p}}_{ij}$	$\boldsymbol{\Omega}^{\mathrm{n}}_{ij}$	$\boldsymbol{\Omega}^{\tau}_{ij}$
planar	$\mathbf{R}_{n_i}\mathrm{diag}(\frac{1}{\epsilon}, 1, 1)\mathbf{R}^T_{n_i}$	$\mathbf{R}_{n_i}\mathrm{diag}(\frac{1}{\epsilon}, \frac{1}{\epsilon}, 1)\mathbf{R}^T_{n_i}$	0
edge	$\mathbf{R}_{\tau_i}\mathrm{diag}(\frac{1}{\epsilon}, \frac{1}{\epsilon}, 1)\mathbf{R}^T_{\tau_i}$	0	$\mathbf{R}_{\tau_i}\mathrm{diag}(\frac{1}{\epsilon}, \frac{1}{\epsilon}, 1)\mathbf{R}^T_{\tau_i}$
none	$\mathbf{I}$	0	0

According to the new formalism, the objective function in Eq. 6 becomes

$$\mathbf{T}^* = \underset{\mathbf{T}}{\mathrm{argmin}} \sum_{\mathcal{C}} \left(\tilde{\mathbf{p}}^{\mathrm{c}}_i - \mathbf{T} \oplus \tilde{\mathbf{p}}^{\mathrm{r}}_j\right)^T \tilde{\boldsymbol{\Omega}}_{ij} \underbrace{\left(\tilde{\mathbf{p}}^{\mathrm{c}}_i - \mathbf{T} \oplus \tilde{\mathbf{p}}^{\mathrm{r}}_j\right)}_{\tilde{\mathbf{e}}_{ij}(\mathbf{T})} , \tag{8}$$

where $\tilde{\boldsymbol{\Omega}}_{ij} = \mathrm{diag}(\boldsymbol{\Omega}^{\mathrm{p}}_{ij}, \boldsymbol{\Omega}^{\mathrm{l}}_{ij})$, and the information matrices must be modified according to Table 2

Table 2: This table summarizes the components of the information matrix for our algorithm when using a reduced representation.

Case	l_i	$\boldsymbol{\Omega}^{\mathrm{p}}_{ij}$	$\boldsymbol{\Omega}^{\mathrm{l}}_{ij}$
planar	$\mathbf{n}_i$	$\mathbf{R}_{n_i}\mathrm{diag}(\frac{1}{\epsilon}, 1, 1)\mathbf{R}^T_{n_i}$	$\mathbf{R}_{n_i}\mathrm{diag}(\frac{1}{\epsilon}, \frac{1}{\epsilon}, 1)\mathbf{R}^T_{n_i}$
edge	$\boldsymbol{\tau}_i$	$\mathbf{R}_{\tau_i}\mathrm{diag}(\frac{1}{\epsilon}, \frac{1}{\epsilon}, 1)\mathbf{R}^T_{\tau_i}$	$\mathbf{R}_{\tau_i}\mathrm{diag}(\frac{1}{\epsilon}, \frac{1}{\epsilon}, 1)\mathbf{R}^T_{\tau_i}$
none	0	$\mathbf{I}$	0

3.2 Carrying on the Optimization

In this section we present the procedure for the minimization described in Eq. 8 by using a strengthened least squares procedure. The input of this algorithm are two sets of extended points $\tilde{\mathbf{p}}^{\mathrm{c}}_{1:n}$ and $\tilde{\mathbf{p}}^{\mathrm{c}}_{1:m}$, a (noisy) set of candidate correspondences $\mathcal{C} = \langle i, j \rangle_{1:M}$ and the information matrix $\tilde{\boldsymbol{\Omega}}_{ij}$, computed according to Table 2. The aim of this procedure is to find the transform $\mathbf{T}^*$ that minimizes the following objective or cost function

$$\mathbf{T}^* = \underset{\mathbf{T}}{\mathrm{argmin}} \sum_{\mathcal{C}} \underbrace{\tilde{\mathbf{e}}_{ij}\left(\mathbf{T}\right)^T \tilde{\boldsymbol{\Omega}}_{ij} \tilde{\mathbf{e}}_{ij}\left(\mathbf{T}\right)}_{\tilde{\chi}^2_{ij}} . \tag{9}$$

Each correspondence contributes to the overall cost function by the scalar term $\tilde{\chi}^2_{ij}$.

As is well know, the minimizing $\mathbf{T}^*$ of Eq. 9 can be found by iteratively solving the following linear system:

$$\mathbf{H} \cdot \mathbf{\Delta T} = -\mathbf{b} \ , \tag{10}$$

where $\mathbf{H} = \sum_{\mathcal{C}} \mathbf{J}_{ij}^T \tilde{\mathbf{\Omega}}_{ij} \mathbf{J}_{ij}$ is the Hessian matrix, $\mathbf{b} = \sum_{\mathcal{C}} \mathbf{J}_{ij}^T \tilde{\mathbf{\Omega}}_{ij} \tilde{\mathbf{e}}_{ij}$ is the coefficient vector and $\mathbf{J}_{ij}$ is the Jacobian of the error function. At each iteration we compute an improved transform $\mathbf{T}'$ from the previous transform $\mathbf{T}$ by using $\mathbf{H}$ and $\mathbf{b}$. By solving the linear system in Eq. 10 we determine a perturbation $\mathbf{\Delta T}$ which is applied to the previous transform $\mathbf{T}$ in order to reduce the error. The transform $\mathbf{T}'$ of the next iteration is thus computed as

$$\mathbf{T}' = \mathbf{\Delta T} \oplus \mathbf{T}' \ . \tag{11}$$

For readers interested in further details on the derivation of Eq. 10 we suggest the work by Kümmerle et al. [2].

In our approach, a perturbation $\mathbf{\Delta T}$ is defined as a vector composed of two parts $(\mathbf{\Delta t} \ \mathbf{\Delta q})^T$ where $\mathbf{\Delta t} = (\mathit{\Delta} t_x \ \mathit{\Delta} t_y \ \mathit{\Delta} t_z)$ is a translation vector and $\mathbf{\Delta q} = (\mathit{\Delta} q_x \ \mathit{\Delta} q_y \ \mathit{\Delta} q_z)$ is the imaginary part of the normalized quaternion used to represent an incremental rotation. If $\mathbf{\Delta t} = 0$, the perturbation is the 4-by-4 identity matrix. Under this parameterization, the Jacobian $\mathbf{J}_{ij}$ with respect to the local perturbation $\mathbf{\Delta T}$ is computed as

$$\mathbf{J}_{ij} = \left. \frac{\partial \mathbf{e}_{ij} \left(\mathbf{\Delta T} \oplus \mathbf{T}^{(n)} \right)}{\partial \mathbf{\Delta T}} \right|_{\mathbf{\Delta T} = \mathbf{0}} = \begin{pmatrix} -\mathbf{I} & 2[\mathbf{T} \oplus \mathbf{p}_j^{\mathrm{r}}]_\times \\ \mathbf{0} & 2[\mathbf{T} \oplus \mathbf{l}_j^{\mathrm{r}}]_\times \end{pmatrix} \ , \tag{12}$$

where $[\mathbf{x}]_\times$ is the cross product matrix of the vector $\mathbf{x}$. In practice, by exploiting the block structure and the sparsity of the Jacobian, it is possible to compute efficiently the linear system in Eq. 10.

In order to be robust to the presence of outliers, which usually significantly contribute to the error, the proposed scheme has to be further modified. To reduce the contribution of these wrong terms, we scale the information matrix of each correspondence whose χ^2 is greater than an acceptance threshold by a factor γ_{ij}.

$$\gamma_{ij} = \begin{cases} 1 & \text{if } \chi_{ij}^2 < K \\ \frac{K}{\chi^2} & \text{otherwise} \end{cases} \tag{13}$$

Even if the correct correspondences are rejected at the beginning of the iterative process, these will be considered again as the system converges towards a better solution, since their χ^2 will decrease.

In order to smooth the convergence it can be also added a damping factor to the linear system in Eq. 10. In practice, $\mathbf{\Delta T}$ is found by solving the damped linear system $(\mathbf{H} + \lambda \mathbf{I}) \mathbf{\Delta T} = -\mathbf{b}$, since it prevents the solution to take too large steps that might be caused by nonlinearities or wrong correspondences.

3.3 Optimization Summary

In this subsection we wrap-up the ideas discussed above and we provide the sketch of an iterative algorithm the optimization of Eq. 8. At each iteration our algorithm computes an improved estimate $\mathbf{T}'$ from the current estimate $\mathbf{T}$ by executing the following steps:

1. Compute the information matrices $\tilde{\mathbf{\Omega}}_{ij}$ according to Table 2;
2. Compute the error vector $\tilde{\mathbf{e}}_{ij}$ as shown in Eq. 8;
3. Compute the Jacobian $\mathbf{J}_{ij}$ according to Eq. 12;
4. Compute the $\tilde{\chi}^2_{ij}$ as in Eq. 9 and the scaling factor γ_{ij} from Eq. 13;
5. Compute a scaled version of the Hessian and of the coefficient vector as:

$$\mathbf{H} = \sum_{\mathcal{C}} \gamma_{ij} \mathbf{J}_{ij}^T \tilde{\mathbf{\Omega}}_{ij} \mathbf{J}_{ij} \qquad \mathbf{b} = \sum_{\mathcal{C}} \gamma_{ij} \mathbf{J}_{ij}^T \tilde{\mathbf{\Omega}}_{ij} \tilde{\mathbf{e}}_{ij} \ ; \tag{14}$$

6. Solve the linear system of Eq. 10 to find an improved perturbation $\mathbf{\Delta T}$;
7. Compute the improved transformation $\mathbf{T}'$ as in Eq. 11.

4 Experiments

We validated our approach both on real and synthetic data. The real world experiments were conducted on publicly available benchmarking datasets, and they show the performances of our optimization algorithm when included in a full ICP system. The experiments on synthetic data, instead, allow to characterize the behavior of our approach under different levels of sensor noise and outlier ratios. Comparisons with NDT are not showed since it performs similarly to GICP because they rely on analogous representations of the points.

4.1 Real World Experiments

For the real world experiments we used the benchmarking datasets by Stuerm *et al.* [7]. Each dataset consists in a sequence of depth and RGB images acquired with a calibrated RGBD camera in a reference scene. Note that even if our approach is not restricted to the use of depth images, we decided to use these datasets since they are labeled with the ground truth of the transformations. We do not make use of the RGB channels.

In order to provide the input data to the algorithm illustrated in the previous section, we processed the point cloud $\mathcal{P}$ generated from each depth image by extracting the local surface characteristics from the neighborhood of each point $\mathbf{p}_i$. This process is performed by computing the parameters of a Gaussian $\mathcal{N}(\mu_i, \mathbf{\Sigma}_i)$ and taking all points that lie within a fixed ball centered in $\mathbf{p}_i$, as

$$\mu_i = \frac{1}{|\mathcal{P}_i|} \sum_{\mathbf{p}_k \in \mathcal{P}_i} \mathbf{p}_k \qquad \mathbf{\Sigma}_i = \frac{1}{|\mathcal{P}_i|} \sum_{\mathbf{p}_k \in \mathcal{P}_i} (\mathbf{p}_k - \mu_i)^T (\mathbf{p}_k - \mu_i) \ , \tag{15}$$

where $\mathcal{P}_i$ is the set of all points in $\mathcal{P}$ that are closer than a fixed distance from $\mathbf{p}_i$.

For determining if a point lies on a corner, an edge or a flat surface, we analyze the eigenvalues of its covariance matrix $\mathbf{\Sigma}_i$. If all eigenvalues have more or less the same magnitude, we assume the point is on a corner. If one of the eigenvalues is smaller with respect to the other two, we assume the point lies on an edge. Finally, if one of the eigenvalues is smaller of some order of magnitude with respect to the others then we assume the point is on a planar patch. This discrimination is necessary to compute the correct information matrices, according to Table 2.

Given two clouds to be aligned, we search the correspondences using a line of sight criterion over the depth images, we reject the correspondences whose normals are too different and we execute one iteration of optimization. Notice that ICP and GICP are special cases that can be captured by our algorithm just by modifying the way in which the information matrices are computed. To focus our analysis on the objective function we left all parts of the system unchanged, including the correspondence selection. This represents an advantage for plain ICP, since normally it does not rely on the normals in order to reject wrong associations.

For each dataset, we incrementally aligned one frame to the previous one. For each iteration of the alignment, we compared the difference between the current solution and the ground truth. Each attempted alignment produced a plot which shows the evolution of the rotational and translational error. For compactness, we provide in this paper only the average error plots obtained by averaging all errors of a run[1]. The reader who is interested in the individual plots of each alignment, can find them at `http://www.dis.uniroma1.it/~serafin/publications/icp-augmented-measurements`.

In order to measure the robustness of the alignments to wrong initial guesses we performed several runs of the experiments by considering a frame each N. Table 3 shows the average evolution of the rotational and the translational error on three different datasets and at different frame skips.

The plotted results point out that our novel objective function in general performs better than the other approaches, in particular in terms of convergence speed. This is true especially for the rotational part of the error since it is influenced directly by the normals. Also in the case where no frame was skipped (first column of Table 3), GICP required twice the number of iterations to converge to the results of our approach. Moreover, ICP and GICP showed much less robustness to frame skipping (second and third column of Table 3).

4.2 Experiments on Synthetic Data

We conducted experiments on synthetic data in order to assess the effects of the inliers and of the sensor error on our optimization function. To this end we generated a scene consisting of about 300k 3D points with normals and tangents. Then, we computed the correct correspondences and ideal measurements and we corrupted them. This process has been performed by injecting a variable fraction of random outliers and perturbing the measurements by adding Gaussian noise

[1] With run we denote all the alignment over a single dataset with a certain frame skip rate.

Table 3: Average evolution of the translational and rotational error for three different datasets at varying frame skip rates. Our approach is labeled "nicp" in the captions of the images.

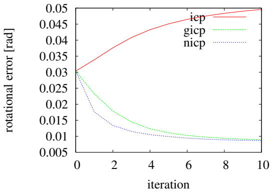
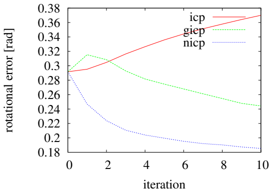
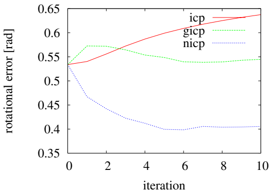
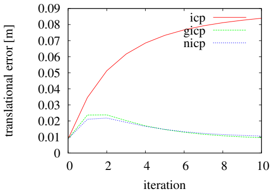
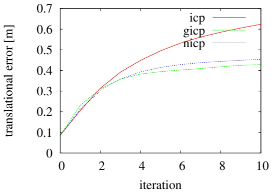
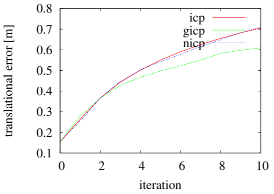
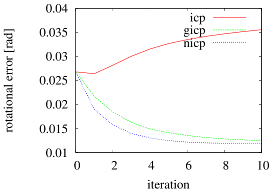
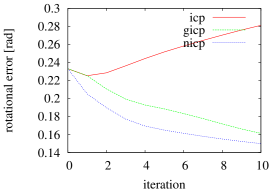
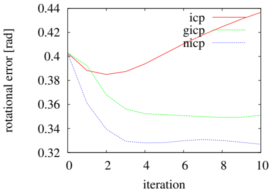
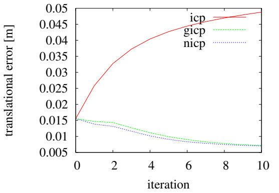
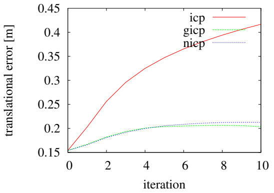
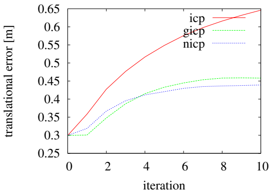
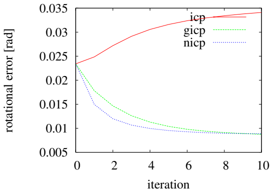
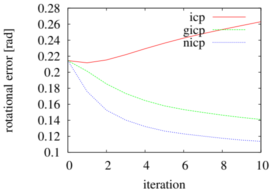
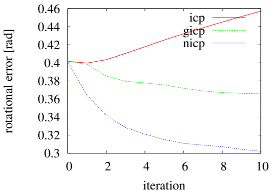
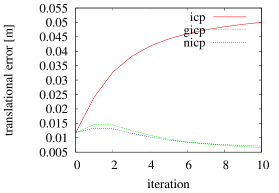
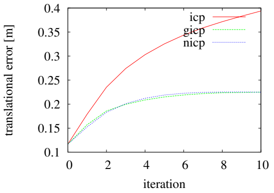
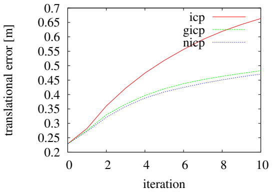
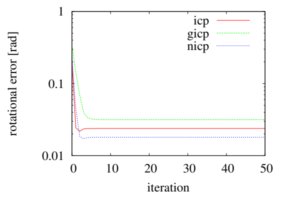
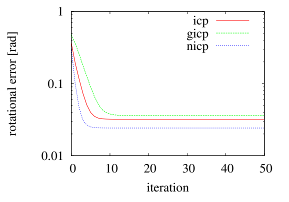
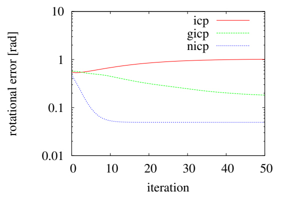
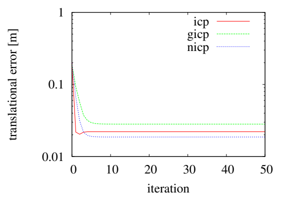
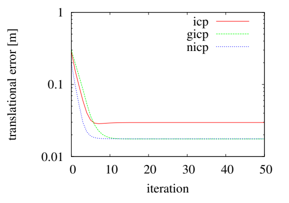
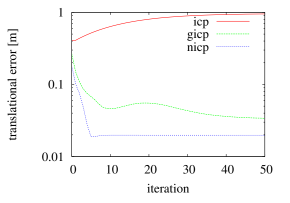
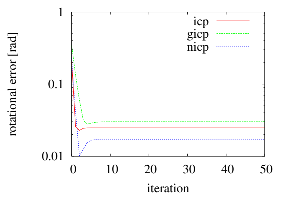
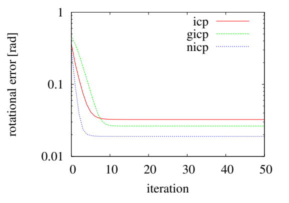
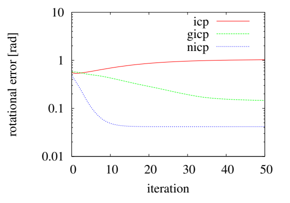
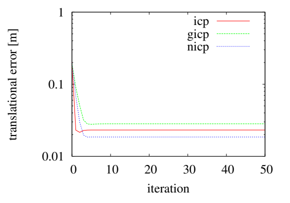
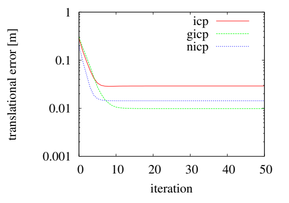
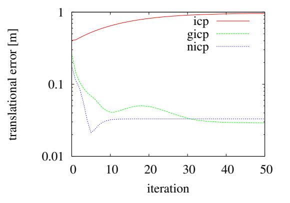
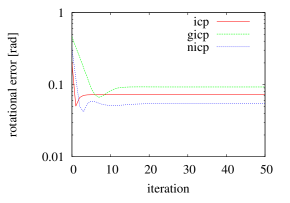
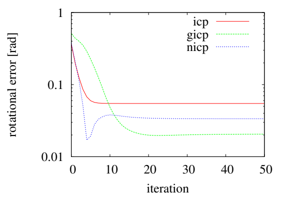
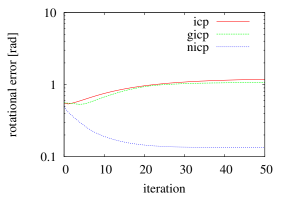
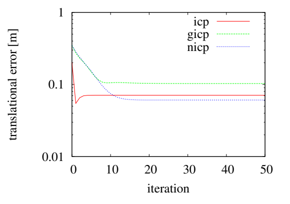
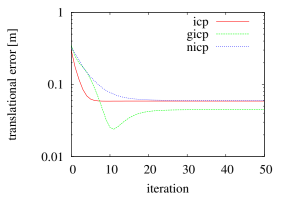
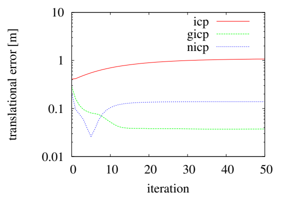
	frame skip rate: 1	frame skip rate: 10	frame skip rate: 20
dataset: fr1/360			
			
dataset: fr1/desk2			
			
dataset: fr1/room			
			

Table 4: Average evolution of the translational and rotational error at different outlier ratios and levels of noise affecting the measurements of the point (standard deviation, in meters) and the normals (standard deviation in degrees). Our approach is labeled "nicp" in the captions of the images.

	outliers: 10%	outliers: 50%	outliers: 90%
noise: 0.1[m], 0°			
			
noise: 0.1[m], 5°			
			
noise: 0.3[m], 20°			
			

to the points and normal estimates. For each setting we ran our approach, ICP and GICP, and we plotted the evolution of the translational and rotational error. The results are shown in Table 4.

Overall the experiments on synthetic data reflect the behavior of the real world ones. Shortly, using additional information in the error function makes the approach more robust and accelerates the convergence. This is particularly true at high rates of outliers and sensor noise. Not surprisingly, instead, noise in the normals lowers the performances. In the unrealistic scenario in which every normal is affected by a 20° error at 90% of outliers the translational estimate becomes less accurate than GICP, but it still converges to a reasonable solution.

5 Conclusions

In this paper we proposed a novel optimization function to register point clouds using an ICP based algorithm that takes into account an augmented measurement vector. Statistical comparative experiments on real and synthetic data show that our approach performs better than other state of the art methods both in terms of convergence speed and robustness. As expected, the normals and the tangents of the surfaces showed an improvement in particular in the rotational part of the error, while keeping the translational one similar to the other approaches. A further enhancement could be obtained by finding an additional measurement, related to the translation, to be considered in the minimization of the cost function.

Acknowledgments. This work has partly been supported by the European Commission under FP7-600890-ROVINA.

References

1. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence (1992)
2. Grisetti, G., Kummerle, R., Stachniss, C., Burgard, W.: A tutorial on graph-based slam. Intelligent Transportation Systems Magazine, IEEE 2(4), 31–43 (2010)
3. Horn, B.K., Hilden, H.M., Negahdaripour, S.: Closed-form solution of absolute orientation using orthonormal matrices. Journal of the Optical Society of America (1988)
4. Magnusson, M., Duckett, T., Lilienthal, A.J.: Scan registration for autonomous mining vehicles using 3d-ndt. Journal on Field Robotics (2007)
5. Pomerleau, F., Colas, F., Siegwart, R., Magnenat, S.: Comparing icp variants on real-world data sets. Autonomous Robots (2013)
6. Segal, A.V., Haehnel, D., Thrun, S.: Generalized-ICP. In: Proc. of Robotics: Science and Systems (RSS) (2009)
7. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS) (2012)