mathematics

Article

Coarse Point Cloud Registration Based on Variational Functionals

Artyom Makovetskii [1,*], Sergei Voronin [1], Vitaly Kober [1,2] and Alexei Voronin [1]

[1] Department of Mathematics, Chelyabinsk State University, 454001 Chelyabinsk, Russia
[2] Department of Computer Science, Ensenada Center for Scientific Research and Higher Education, Ensenada 22860, BC, Mexico
* Correspondence: artemmac@csu.ru

Abstract: Point cloud collection forming a 3D scene typically uses information from multiple data scans. The common approach is to register the point cloud pairs consequentially using a variant of the iterative closest point (ICP) algorithm, but most versions of the ICP algorithm only work correctly for a small movement between two point clouds. This makes it difficult to accumulate multiple scans. Global registration algorithms are also known, which theoretically process point clouds at arbitrary initial positions. Recently, a multiparameter variational functional was described and used in the ICP variant to register point clouds at arbitrary initial positions. The disadvantage of this algorithm was the need for manual selection of parameters. In this paper, a modified version of the algorithm with automatic selection of the model parameters is proposed. The proposed algorithm is a fusion of the ICP and RANSAC concepts. Moreover, the algorithm can be parallelized. The performance of the proposed algorithm is compared with that of known global registration algorithms.

Keywords: point cloud registration; variational functional; global registration; coarse registration; closed-form solution; iterative closest points (ICP); orthogonal transformation; surface reconstruction

MSC: 49K99; 68T45

check for updates

Citation: Makovetskii, A.; Voronin, S.; Kober, V.; Voronin, A. Coarse Point Cloud Registration Based on Variational Functionals. *Mathematics* **2023**, *11*, 35. https://doi.org/10.3390/math11010035

Academic Editor: Jakub Nalepa

Received: 20 November 2022
Revised: 12 December 2022
Accepted: 16 December 2022
Published: 22 December 2022

1. Introduction

In computer vision, splicing different parts of 3D models is necessary to reconstruct the objects in three-dimensional space. Point cloud registration is a basic element to solve this problem. Point cloud registration is a widely studied problem. Known algorithms in this area are described in surveys [1,2]. Registration algorithms can be divided into two groups. The first of them contains refinement algorithms, and the second one contains coarse registration algorithms. Refinement algorithms are used to align the source point cloud P and the target point cloud Q if the rotation angles connecting P and Q are small enough. Coarse registration algorithms are used in the case of sufficiently large angles.

The following abbreviations are used throughout the paper: ICP—iterative closest points algorithm; PtP-ICP—point-to-point ICP; PCL-GICP—software implementation of generalized ICP (GICP) from point cloud library; NICP—normal ICP; RANSAC—random sample consensus algorithm; Cluster—registration algorithm based on the cluster approach; Super4PCS—global registration algorithm; Nsamples—overlap–parameters of Super4PCS; λ-ICP—registration algorithm based on the multiparameter λ-functional; λ_r-ICP—registration algorithm based on the proposed λ_r-functional; LCP—largest common pointset; Delta—distance threshold for LCP; SHOT—signature of histograms of orientations descriptor; threshold_PtP—threshold of the distance for corresponding point pairs for PtP-ICP; GL (3)—general linear group in three dimensions; SO(3)—special orthogonal group in three dimensions.

The iterative closest points algorithm (ICP) [3,4] is a common refinement algorithm. The two basic steps of the ICP algorithm alternate with each other, that is, the estimation of a geometric transformation based on a fixed correspondence (variational problem of the ICP algorithm) and updating the correspondences to their closest matches (correspondence step).

The most well-known functionals used in the ICP variational problem are presented in ICP variants, such as point-to-point [3], point-to-plane [4], generalized ICP (GICP) [5], and normal ICP (NICP) [6,7]. Closed-form solutions to the affine point-to-point [8] and point-to-plane [9,10] problems have been proposed. Closed-form solutions to the orthogonal point-to-point problem [11–13] have also been obtained. Generalized ICP [5] and NICP [6,7] with other functionals are more robust. In these cases, iterative conjugate gradients and Gauss–Newton methods [5,6] are used to solve the variational ICP problems.

Stochastic methods based on the grey wolf optimizer (GWO) algorithm are exploited to solve the variational ICP problems. The GWO has recently been used to a coarse point cloud alignment [14] and implementation of the point-to-point ICP algorithm [15].

The paper in [6] presents an experimental comparative analysis of the NICP algorithm, generalized ICP (GICP), NDT [16], and KinFu [17]. The analysis shows that the NICP yields better results and higher robustness compared to the state-of-the-art methods [6]. A closed-form solution to the NICP variational problem was proposed [7]. This solution was compared with the performance of the iterative Gauss–Newton method [6].

Coarse registration approaches usually use some form of the well-known RANSAC algorithm [18] or related methods to confirm a congruent relationship [19–21]. Based on this technique, 4PCS [22] and Super4PCS [23] methods have achieved a significant increase in speed compared to alternative approaches and provide unstructured and efficient scene acquisition at scales that were previously impossible. In addition, feature-based methods such as SHOT [24] and FPFH [25] are also popular; however, they are sensitive to noise. SDRSAC [26] uses a randomized strategy without correspondences.

Registration methods based on neural networks can also be regarded as coarse registration algorithms. Recently, point cloud classification, segmentation, and shape matching have been carried out using neural networks [27–32]. So, to solve the problem of registering point clouds, a neural network referred to as deep closest points (DCP) was designed [33]. DCP is trained on the ModelNet40 3D point clouds database [34].

Recently, the λ-functional and λ-ICP algorithms have been proposed [35]. The λ-functional is a generalized version of the NICP functional. In order to compute the NICP functional, the covariance matrix in the vicinity of each cloud point is calculated. By selecting diagonal elements for a given point cloud, the probability of NICP convergence to acceptable solution can be increased [7]. A similar approach is also used for the λ-functional.

In real applications, a registration algorithm deals with incongruent point clouds. Therefore, the performance of registration algorithms should be compared on incongruent clouds. It was shown in [35] that λ-ICP based on the λ-functional is very efficient, but the drawback of λ-ICP is the need to select the λ parameters manually for a given type of point cloud.

In this paper, a coarse registration algorithm based on a new λ_r-functional, which is a reduced version of the λ-functional, is proposed. The proposed ICP-type algorithm, referred to as λ_r-ICP, contains some elements based on RANSAC and deals with coarse registration of incongruent point clouds. To solve the λ_r-ICP variational problem, first the solution to the variational problem in the class GL (3) is obtained. Next, the result is projected onto the submanifold SO(3). Finally, the translation vector is computed. The variational problem λ-ICP [35], NICP [7] and point-to-plane ICP [36] is solved similarly.

In this paper, we propose a new solution to the first step of the scheme described above. The λ_r-functional is a reduced version of the λ-functional with special properties. In particular, the λ-functional contains terms that are vector projections onto local vector bases, while the λ_r-functional contains terms that are vector projections onto the global vector basis. This requires a different approach to calculating the functional gradient. To minimize the λ-functional, 4 × 4 transformation matrices in homogeneous coordinates were used, while 3 × 3 matrices in Cartesian coordinates were used for the λ_r-functional. The resulting calculation formulas for the λ_r-functional differ significantly from the corresponding formulas for the λ-functional. The sequential (i.e., nonparallel) software implementation of the proposed solution to minimize the λ_r-functional is about 10 times faster than the sequential software implementation for the λ-functional. In addition, the proposed solution calculates three rows of the rotation matrix independently, in parallel computational threads.

Note that the λ-functional and the λ_r-functional contain, in a certain sense, the point-to-point, point-to-plane, and NICP functionals. The parameters λ_1, λ_2, and λ_3 of the λ_r-functional are computed by averaging the local geometric information of all points in the point cloud. After that, two free parameters, λ_4 and k, where k is the neighborhood size, are used. The λ_r-ICP algorithm contains iterating over two values of k and sixteen values of λ_4, i.e., the algorithm forms 32 geometric transformations. The choice of the best transformation is carried out using the largest common pointset (LCP) [22,23]. Note that the proposed λ_r-ICP algorithm contains elements of both ICP and RANSAC approaches.

The proposed algorithm is compared with known coarse registration algorithms, Super4PCS and RANSAC, using the software implementation of Super4PCS [37]. The program generates coarse geometric transformations and returns the best result according to the LCP criterion. After that, we apply refinement with the point-to-point ICP with the selection of pairs of corresponding points.

The λ_r-ICP and RANSAC algorithms are organized in the following way. First, these algorithms form a set of coarse geometric transformations. Second, refinements are applied to each coarse transformation. Third, the best transformation is selected according to the LCP criterion.

We also use the following algorithms for comparison: point-to-point ICP with selection of pairs of corresponding points; GICP [5] (its software implementation from the point cloud library (PCL) [38]); and coarse registration algorithm based on the cluster approach [39,40].

It is known that conventional eigenvalue decomposition algorithms produce eigenvectors with unpredictable directions [24,41]. The accurate choice of orientation of the eigenvectors is important for the correct operation of the λ_r-ICP algorithm. In [35], an algorithm was proposed for reorienting the eigenvectors calculated in a neighborhood inside a point cloud. In this paper, a slightly modified version of this algorithm is described. The modified algorithm is called orientation predictor and is based on a local geometric descriptor. The descriptor is also used for selecting pairs of corresponding points in the λ_r-ICP algorithm.

The main contributions of the proposed paper are as follows:

1. A new λ_r-functional is proposed, which is a reduced version of the previously described λ-functional;
2. A new computational approach to minimization of the λ_r-functional is proposed. This approach essentially differs from the corresponding one for the λ-functional and is much faster;
3. A new registration algorithm λ_r-ICP is proposed, which automatically computes the parameters of the λ_r-functional for the given source and target incongruent noisy point clouds. The λ_r-ICP algorithm is a fusion of the ICP and RANSAC concepts.

The paper is organized as follows. Section 2 proposes an exact solution approximation to the variation problem with the λ_r-functional. In Section 3, the orientation predictor and λ_r-ICP algorithm are described. Section 4 presents and discusses the results of computer simulations. Section 5 summarizes our findings.

2. Reduced λ-Functional

The proposed approach is based on the λ-functional described in [35]. In [35], the values of the λ-functional parameters are selected manually. Here we propose an approach to automatic parameters search.

2.1. Local Geometric Features

Let P and Q be the source and target point clouds, respectively, with a given correspondence between them (from the source cloud P to the target cloud Q), $P = \{p_1, \dots p_s\}$ and $Q = \{q_1, \dots q_s\}$, s is the number of points in the cloud P. Note that we consider clouds P and Q with an already given correspondence, the initial quantity of points in P and Q may be different. Let us consider the k-neighborhood $\{q_{i1}, \dots, q_{ik}\}$ of the point q_i, $i = 1, \dots, s$, in the cloud Q. Denoted by c_i, the mass center of the neighborhood, $c_i = \frac{1}{k}\sum_{j=1}^{k} q_{ij}$. The covariance matrix Σ_i of the neighborhood is defined as

$$\Sigma_i = \widetilde{Q}\widetilde{Q}^t \tag{1}$$

where

$$\widetilde{Q} = \begin{pmatrix} \widetilde{q}_{i11} & & \widetilde{q}_{ik1} \\ \widetilde{q}_{i12} & \cdots & \widetilde{q}_{ik2} \\ \widetilde{q}_{i13} & & \widetilde{q}_{ik3} \end{pmatrix}$$

$$\widetilde{q}_{ij} = q_{ij} - c_i,\ j = 1, \dots, k$$

The eigendecomposition of the matrix Σ_i can be written as

$$\Sigma_i = \Re_i \Lambda_i {\Re_i}^t \tag{2}$$

$$\Lambda_i = \begin{pmatrix} \lambda_{i1} & 0 & 0 \\ 0 & \lambda_{i2} & 0 \\ 0 & 0 & \lambda_{i3} \end{pmatrix}$$

where $\lambda_{i3} \geq \lambda_{i2} \geq \lambda_{i1} > 0$ are the eigenvalues of the covariance matrix, and $\Re_i$ is the orthogonal matrix of the eigenvectors. The matrix $\Re_i$ consists of the vector-columns r_{i1}, r_{i2}, and r_{i3}, corresponding to the eigenvalues λ_{i1}, λ_{i2}, and λ_{i3}. The vector r_{i1} can be considered as a normal vector to the tangent plane at the point q_{ij}, and the vector r_{i3} corresponds to the main axis of the inertia ellipsoid of the given neighborhood.

The covariance matrix for the point cloud P is defined in a similar manner.

2.2. λ-Functional and Its Variants

The λ-functional is given as follows [35]:

$$\begin{gathered} J_\lambda(R,T) = \sum_{i=1}^{s} \left(R{r_{i1}}^p - {r_{i1}}^q\right)^t {\Re_i}^q {\Lambda_{i1}}^q \left({\Re_i}^q\right)^t \left(R{r_{i1}}^p - {r_{i1}}^q\right) + \left(R{r_{i2}}^p - {r_{i2}}^q\right)^t {\Re_i}^q {\Lambda_{i2}}^q \left({\Re_i}^q\right)^t \left(R{r_{i2}}^p - {r_{i2}}^q\right) + \\ \left(R{r_{i3}}^p - {r_{i3}}^q\right)^t {\Re_i}^q {\Lambda_{i3}}^q \left({\Re_i}^q\right)^t \left(R{r_{i3}}^p - {r_{i3}}^q\right) + \\ \left(Rp_i + T - q_i\right)^t {\Re_i}^q {\Lambda_{i4}}^q \left({\Re_i}^q\right)^t \left(Rp_i + T - q_i\right) + \\ \left(R^t{\rho_{i1}}^q - {\rho_{i1}}^p\right)^t {\Re_i}^p {\Lambda_{i1}}^p \left({\Re_i}^p\right)^t \left(R^t{\rho_{i1}}^q - {\rho_{i1}}^p\right) + \left(R^t{\rho_{i2}}^q - {\rho_{i2}}^p\right)^t {\Re_i}^p {\Lambda_{i2}}^p \left({\Re_i}^p\right)^t \left(R^t{\rho_{i2}}^q - {\rho_{i2}}^p\right) + \\ \left(R^t{\rho_{i3}}^q - {\rho_{i3}}^p\right)^t {\Re_i}^p {\Lambda_{i3}}^p \left({\Re_i}^p\right)^t \left(R^t{\rho_{i3}}^q - {\rho_{i3}}^p\right) \end{gathered} \tag{3}$$

where ${r_{i1}}^p$, ${r_{i2}}^p$, and ${r_{i3}}^p$ are vector columns of the orthogonal matrix ${\Re_i}^p$ (of point cloud P), ${\Lambda_{i1}}^q$, ${\Lambda_{i2}}^q$, and ${\Lambda_{i3}}^q$ are diagonal matrices with elements $\left({\lambda_{ij1}}^q, {\lambda_{ij1}}^q, {\lambda_{ij1}}^q\right)$, $j - 1, 2, 3$, which are considered variable parameters, and the vectors ${r_{i1}}^q$, ${r_{i2}}^q$, and ${r_{i3}}^q$ are vector columns of the orthogonal matrix ${\Re_i}^q$ (of point cloud Q). The elements of the diagonal matrix ${\Lambda_{i4}}^q$ are also considered variable parameters. The vectors ${\rho_{i1}}^q$, ${\rho_{i2}}^q$, and ${\rho_{i3}}^q$, the vectors ${\rho_{i1}}^p$, ${\rho_{i2}}^p$, and ${\rho_{i3}}^q$, and the matrices ${\Lambda_{i1}}^p$, ${\Lambda_{i2}}^p$, and ${\Lambda_{i3}}^p$ are defined similarly for the inverse correspondence from P to Q. R is the rotation matrix and T is the translation vector.

The ICP variational problem (error minimization ICP step) based on the λ-functional can be written as

$$(R_*, T_*) = arg\min_{R,T} J_\lambda(R,T) \tag{4}$$

subject to $R_* \in SO(3)$. The ICP-type algorithm with the variational problem (4) is called λ-ICP. In [35], manual selction of parameters was considered, i.e., elements of the diagonal matrices ${\Lambda_{i1}}^q$, ${\Lambda_{i2}}^q$, ${\Lambda_{i3}}^q$, ${\Lambda_{i4}}^q$, ${\Lambda_{i1}}^p$, ${\Lambda_{i2}}^p$, and ${\Lambda_{i3}}^p$ for a given point cloud pair (P,Q). In this paper, we propose an approach to find the parameters of the λ-functional automatically.

The first term in (3) can be rewritten as

$$({Rr_{i1}}^p - {r_{i1}}^q)^t {\Re_i}^q {\Lambda_{i1}}^q ({\Re_i}^q)^t ({Rr_{i1}}^p - {r_{i1}}^q) = {\lambda_{i11}}^q \langle {r_{i1}}^q, R{r_{i1}}^p - {r_{i1}}^q \rangle^2 + {\lambda_{i12}}^q \langle {r_{i2}}^q, R{r_{i1}}^p - {r_{i1}}^q \rangle^2 + \\ {\lambda_{i13}}^q \langle {r_{i3}}^q, R{r_{i1}}^p - {r_{i1}}^q \rangle^2 \tag{5}$$

where ${\Lambda_{i1}}^q = diag({\lambda_{i11}}^q, {\lambda_{i12}}^q, {\lambda_{i13}}^q)$ and ${\Re_i}^q = ({r_{i1}}^q\ {r_{i2}}^q\ {r_{i3}}^q)$. It follows from (5) that the four first terms in (3) take the form

$$\begin{gathered}\sum_{i=1}^{s} (R{r_{i1}}^p - {r_{i1}}^q)^t {\Re_i}^q {\Lambda_{i1}}^q ({\Re_i}^q)^t (R{r_{i1}}^p - {r_{i1}}^q) + (R{r_{i2}}^p - {r_{i2}}^q)^t {\Re_i}^q {\Lambda_{i2}}^q ({\Re_i}^q)^t (R{r_{i2}}^p - {r_{i2}}^q) + \\ (R{r_{i3}}^p - {r_{i3}}^q)^t {\Re_i}^q {\Lambda_{i3}}^q ({\Re_i}^q)^t (R{r_{i3}}^p - {r_{i3}}^q) + \\ (Rp_i + T - q_i)^t {\Re_i}^q {\Lambda_{i4}}^q ({\Re_i}^q)^t (Rp_i + T - q_i) = \\ \sum_{i=1}^{s} {\lambda_{i11}}^q \langle {r_{i1}}^q, R{r_{i1}}^p - {r_{i1}}^q \rangle^2 + {\lambda_{i12}}^q \langle {r_{i2}}^q, R{r_{i1}}^p - {r_{i1}}^q \rangle^2 + {\lambda_{i13}}^q \langle {r_{i3}}^q, R{r_{i1}}^p - {r_{i1}}^q \rangle^2 + \\ {\lambda_{i21}}^q \langle {r_{i1}}^q, R{r_{i2}}^p - {r_{i2}}^q \rangle^2 + {\lambda_{i22}}^q \langle {r_{i2}}^q, R{r_{i2}}^p - {r_{i2}}^q \rangle^2 + {\lambda_{i23}}^q \langle {r_{i3}}^q, R{r_{i2}}^p - {r_{i2}}^q \rangle^2 + \\ {\lambda_{i31}}^q \langle {r_{i1}}^q, R{r_{i3}}^p - {r_{i3}}^q \rangle^2 + {\lambda_{i32}}^q \langle {r_{i2}}^q, R{r_{i3}}^p - {r_{i3}}^q \rangle^2 + {\lambda_{i33}}^q \langle {r_{i3}}^q, R{r_{i3}}^p - {r_{i3}}^q \rangle^2 + \\ {\lambda_{i4}}^q (Rp_i + T - q_i)^t (Rp_i + T - q_i)\end{gathered} \tag{6}$$

where ${\Lambda_{i4}}^q = diag({\lambda_{i4}}^q, {\lambda_{i4}}^q, {\lambda_{i4}}^q)$.

2.3. Relationship between λ-Functional and Point-to-Point, Point-to-Plane, and NICP Functionals

Let us consider the relationship between such well-known ICP functionals as point-to-point [3], point-to-plane [4], NICP [6,7], and λ-functional [35]. Note that the tens term of the right side of (6) can be written as

$$\sum_{i=1}^{s} {\lambda_{i4}}^q (Rp_i + T - q_i)^t (Rp_i + T - q_i) = \sum_{i=1}^{s} {\lambda_{i4}}^q ||Rp_i + T - q_i||^2 \tag{7}$$

The functional in (7) is a weighted point-to-point functional.

The first term of the right side of (6) is given as

$$\sum_{i=1}^{s} {\lambda_{i11}}^q \langle {r_{i1}}^q, R{r_{i1}}^p - {r_{i1}}^q \rangle^2 \tag{8}$$

Since the vector ${r_{i1}}^q$ can be considered normal to the tangent plane at the point q_i, the expression (8) is mathematically close to the weighted point-to-plane functional.

The first and fourth terms on the left side of (6) are the NICP functional and given as

$$\sum_{i=1}^{s} (R{r_{i1}}^p - {r_{i1}}^q)^t {\Re_i}^q {\Lambda_{i1}}^q ({\Re_i}^q)^t (R{r_{i1}}^p - {r_{i1}}^q) + (Rp_i + T - q_i)^t {\Re_i}^q {\Lambda_{i4}}^q ({\Re_i}^q)^t (Rp_i + T - q_i) \tag{9}$$

Therefore, the λ-functional is a generalization of the point-to-point, point-to-plane, and NICP functionals.

2.4. λ_r-Functional

Let us describe a reduced version of the λ-functional, referred to as λ_r-functional. Consider the following functionals:

$$J_{11}(R) = \lambda_1 \sum_{i=1}^{s} \langle e_1, R{r_{i1}}^p - {r_{i1}}^q \rangle^2, J_{12}(R) = \lambda_2 \sum_{i=1}^{s} \langle e_2, R{r_{i1}}^p - {r_{i1}}^q \rangle^2, J_{13}(R) = \lambda_3 \sum_{i=1}^{s} \langle e_3, R{r_{i1}}^p - {r_{i1}}^q \rangle^2$$

$$J_{21}(R) = \lambda_1 \sum_{i=1}^{s} \langle e_1, Rr_{i2}{}^{p} - r_{i2}{}^{q} \rangle^2, J_{22}(R) = \lambda_2 \sum_{i=1}^{s} \langle e_2, Rr_{i2}{}^{p} - r_{i2}{}^{q} \rangle^2, J_{23}(R) = \lambda_3 \sum_{i=1}^{s} \langle e_3, Rr_{i2}{}^{p} - r_{i2}{}^{q} \rangle^2$$

$$J_{31}(R) = \lambda_1 \sum_{i=1}^{s} \langle e_1, Rr_{i3}{}^{p} - r_{i3}{}^{q} \rangle^2, J_{32}(R) = \lambda_2 \sum_{i=1}^{s} \langle e_2 Rr_{i3}{}^{p} - r_{i3}{}^{q} \rangle^2, J_{33}(R) = \lambda_3 \sum_{i=1}^{s} \langle e_3, Rr_{i3}{}^{p} - r_{i3}{}^{q} \rangle^2$$

$$J_4(R,T) = \lambda_4 \sum_{i=1}^{s} \langle Rp_i + T - q_i, Rp_i + T - q_i \rangle^2 \tag{10}$$

where $e_1 = (1\ 0\ 0)^t$, $e_2 = (0\ 1\ 0)^t$, and $e_3 = (0\ 0\ 1)^t$ are basis vectors of the coordinate system (in which clouds P and Q are considered). All coefficients λ ($\lambda > 0$) of (10) do not depend on the number of points.

Definition 1. *The following functional is called the λ_r-functional:*

$$J_{\lambda\_\mathrm{r}}(R,T) = J_{11}(R) + J_{12}(R) + J_{13}(R) + J_{21}(R) + J_{22}(R) + J_{23}(R) + J_{31}(R) + J_{32}(R) + J_{33}(R) + J_4(R,T) \tag{11}$$

The λ_r-ICP variational problem can be written as:

$$(R_*, T_*) = arg\min_{R,T} J_{\lambda\_\mathrm{r}}(R,T) \tag{12}$$

subject to $R_* \in SO(3)$.

The λ_r-functional is obtained from the λ-functional by reduction. Suppose that $\Re_i{}^q = I$, and $\Lambda_{i1}{}^q = diag(\lambda_1{}^q, \lambda_1{}^q, \lambda_1{}^q)$, $\Lambda_{i2}{}^q = diag(\lambda_2{}^q, \lambda_2{}^q, \lambda_2{}^q)$, $\Lambda_{i3}{}^q = diag(\lambda_3{}^q, \lambda_3{}^q, \lambda_3{}^q)$, $\Lambda_4{}^q = diag(\lambda_4{}^q, \lambda_4{}^q, \lambda_4{}^q)$, $i = 1, \ldots, s$. Note that information from the matrices $\Re_i{}^q$ is used in expressions such as $(Rr_{i1}{}^p - r_{i1}{}^q)$.

2.5. Gradient of the λ_r-Functional

c_p and c_q denote the mass centers of the point clouds P and Q, respectively. Let us $P\prime$ and $Q\prime$ also denote the following point clouds:

$$P' = \{p_1', \ldots p_s'\} = \{p_1 - c_p, \ldots p_s - c_p\},\ Q' = \{q_1', \ldots q_s'\} = \{q_1 - c_q, \ldots q_s - c_q\} \tag{13}$$

The λ_r-functional for clouds P' and Q' takes the following form:

$$J_{\lambda\_\mathrm{r}}(R,T) = J_{11}(R) + J_{12}(R) + J_{13}(R) + J_{21}(R) + J_{22}(R) + J_{23}(R) + J_{31}(R) + J_{32}(R) + J_{33}(R) + J_{4\prime}(R) \tag{14}$$

where

$$J_{4\prime}(R) = \lambda_4 \sum_{i=1}^{s} \langle Rp\prime_i - q\prime_i, Rp\prime_i - q\prime_i \rangle^2 \tag{15}$$

The λ_r-ICP variational problem can be rewritten as

$$R_* = arg\min_{R} J_{\lambda\_\mathrm{r}}(R) \tag{16}$$

subject to $R_* \in SO(3)$.

We have that

$$\nabla J_{11}(R) = 2\lambda_1 \sum_{i=1}^{s} \langle e_1, Rr_{i1}{}^p \rangle e_1 (r_{i1}{}^p)^t - 2\lambda_1 \sum_{i=1}^{s} \langle e_1, r_{i1}{}^q \rangle e_1 (r_{i1}{}^p)^t \tag{17}$$

Let us denote the elements of R and $r_{i1}{}^p$ as

$$R = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix},\ r_{i1}{}^p = (r_{i11}{}^p\ r_{i12}{}^p\ r_{i13}{}^p)^t \tag{18}$$

Note that $\langle e_1, Rr_{i1}{}^p \rangle = \langle e_1 (r_{i1}{}^p)^t, R \rangle$ and

$$\langle e_1, Rr_{i1}{}^p \rangle e_1 (r_{i1}{}^p)^t = \langle e_1 (r_{i1}{}^p)^t, R \rangle e_1 (r_{i1}{}^p)^t =$$

$$\left\langle \begin{pmatrix} {r_{i11}}^{p} & {r_{i12}}^{p} & {r_{i13}}^{p} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, R \right\rangle \begin{pmatrix} {r_{i11}}^{p} & {r_{i12}}^{p} & {r_{i13}}^{p} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} =$$

$$\begin{pmatrix} {r_{i11}}^{p}({r_{i11}}^{p}R_{11} + {r_{i12}}^{p}R_{12} + {r_{i13}}^{p}R_{13}) & {r_{i12}}^{p}({r_{i11}}^{p}R_{11} + {r_{i12}}^{p}R_{12} + {r_{i13}}^{p}R_{13}) & {r_{i13}}^{p}({r_{i11}}^{p}R_{11} + {r_{i12}}^{p}R_{12} + {r_{i13}}^{p}R_{13}) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} =$$

$$\begin{pmatrix} ({r_{i11}}^{p})^2R_{11} + {r_{i11}}^{p}{r_{i12}}^{p}R_{12} + {r_{i11}}^{p}{r_{i13}}^{p}R_{13} & {r_{i12}}^{p}{r_{i11}}^{p}R_{11} + ({r_{i12}}^{p})^2R_{12} + {r_{i12}}^{p}{r_{i13}}^{p}R_{13} & {r_{i13}}^{p}{r_{i11}}^{p}R_{11} + {r_{i13}}^{p}{r_{i12}}^{p}R_{12} + ({r_{i13}}^{p})^2R_{13} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{19}$$

The gradient $\nabla J_{11}(R)$ depends only on the first row of the matrix obtained in (19). This row in transposed form can be rewritten as

$$\begin{pmatrix} ({r_{i11}}^{p})^2 & {r_{i11}}^{p}{r_{i12}}^{p} & {r_{i11}}^{p}{r_{i13}}^{p} \\ {r_{i1}}^{p}{}_{2}{r_{i11}}^{p} & ({r_{i12}}^{p})^2 & {r_{i12}}^{p}{r_{i13}}^{p} \\ {r_{i13}}^{p}{r_{i11}}^{p} & {r_{i1}}^{p}{}_{3}{r_{i12}}^{p} & ({r_{i13}}^{p})^2 \end{pmatrix} \begin{pmatrix} R_{11} \\ R_{12} \\ R_{13} \end{pmatrix} = \left({r_{i1}}^{p}({r_{i1}}^{p})^{t}\right) \begin{pmatrix} R_{11} \\ R_{12} \\ R_{13} \end{pmatrix}. \tag{20}$$

The gradient $(\nabla J_{11}(R))^t$ takes the form:

$$(\nabla J_{11}(R))^t = 2\lambda_1 \textstyle\sum_{i=1}^{s}\left({r_{i2}}^{p}({r_{i2}}^{p})^t\right) \begin{pmatrix} R_{11} \\ R_{12} \\ R_{13} \end{pmatrix} - 2\lambda_1(\sum_{i=1}^{s}\langle e_1, {r_{i2}}^{q}\rangle e_1({r_{i2}}^{p})^t)^t{}_1, \tag{21}$$

where $(\sum_{i=1}^{s}\langle e_1, {r_{i2}}^{q}\rangle e_1({r_{i2}}^{p})^t)^t{}_1$ means the first column of the matrix $\sum_{i=1}^{s}\langle e_1, {r_{i2}}^{q}\rangle e_1({r_{i2}}^{p})^t$.

In a similar manner, the gradients $\nabla J_{12}(R)$, $\nabla J_{13}(R)$, $\nabla J_{21}(R)$, $\nabla J_{22}(R)$, $\nabla J_{23}(R)$, $\nabla J_{31}(R)$, $\nabla J_{32}(R)$, and $\nabla J_{33}(R)$ are computed as

$$(\nabla J_{12}(R))^t = 2\lambda_2 \textstyle\sum_{i=1}^{s}\left({r_{i1}}^{p}({r_{i1}}^{p})^t\right) \begin{pmatrix} R_{21} \\ R_{22} \\ R_{23} \end{pmatrix} - 2\lambda_2(\sum_{i=1}^{s}\langle e_2, {r_{i1}}^{q}\rangle e_2({r_{i1}}^{p})^t)^t{}_2, \tag{22}$$

$$(\nabla J_{13}(R))^t = 2\lambda_3 \textstyle\sum_{i=1}^{s}\left({r_{i1}}^{p}({r_{i1}}^{p})^t\right) \begin{pmatrix} R_{31} \\ R_{32} \\ R_{33} \end{pmatrix} - 2\lambda_3(\sum_{i=1}^{s}\langle e_3, {r_{i1}}^{q}\rangle e_3({r_{i1}}^{p})^t)^t{}_3, \tag{23}$$

$$(\nabla J_{21}(R))^t = 2\lambda_1 \textstyle\sum_{i=1}^{s}\left({r_{i2}}^{p}({r_{i2}}^{p})^t\right) \begin{pmatrix} R_{11} \\ R_{12} \\ R_{13} \end{pmatrix} - 2\lambda_1(\sum_{i=1}^{s}\langle e_1, {r_{i2}}^{q}\rangle e_1({r_{i2}}^{p})^t)^t{}_1, \tag{24}$$

$$(\nabla J_{22}(R))^t = 2\lambda_2 \textstyle\sum_{i=1}^{s}\left({r_{i2}}^{p}({r_{i2}}^{p})^t\right) \begin{pmatrix} R_{21} \\ R_{22} \\ R_{23} \end{pmatrix} - 2\lambda_2(\sum_{i=1}^{s}\langle e_2, {r_{i2}}^{q}\rangle e_2({r_{i2}}^{p})^t)^t{}_2, \tag{25}$$

$$(\nabla J_{23}(R))^t = 2\lambda_3 \textstyle\sum_{i=1}^{s}\left({r_{i2}}^{p}({r_{i2}}^{p})^t\right) \begin{pmatrix} R_{31} \\ R_{32} \\ R_{33} \end{pmatrix} - 2\lambda_3(\sum_{i=1}^{s}\langle e_3, {r_{i2}}^{q}\rangle e_3({r_{i2}}^{p})^t)^t{}_3, \tag{26}$$

$$(\nabla J_{31}(R))^t = 2\lambda_1 \textstyle\sum_{i=1}^{s}\left({r_{i3}}^{p}({r_{i3}}^{p})^t\right) \begin{pmatrix} R_{11} \\ R_{12} \\ R_{13} \end{pmatrix} - 2\lambda_1(\sum_{i=1}^{s}\langle e_1, {r_{i3}}^{q}\rangle e_1({r_{i3}}^{p})^t)^t{}_1, \tag{27}$$

$$(\nabla J_{32}(R))^t = 2\lambda_2 \textstyle\sum_{i=1}^{s}\left({r_{i3}}^{p}({r_{i3}}^{p})^t\right) \begin{pmatrix} R_{21} \\ R_{22} \\ R_{23} \end{pmatrix} - 2\lambda_2(\sum_{i=1}^{s}\langle e_2, {r_{i3}}^{q}\rangle e_2({r_{i3}}^{p})^t)^t{}_2, \tag{28}$$

$$(\nabla J_{33}(R))^t = 2\lambda_3 \textstyle\sum_{i=1}^{s}\left({r_{i3}}^{p}({r_{i3}}^{p})^t\right) \begin{pmatrix} R_{31} \\ R_{32} \\ R_{33} \end{pmatrix} - 2\lambda_3(\sum_{i=1}^{s}\langle e_3, {r_{i3}}^{q}\rangle e_3({r_{i3}}^{p})^t)^t{}_3. \tag{29}$$

Let us compute the gradient $\nabla J_4(R)$

$$(\nabla J_4(R))^t = 2\lambda_4 \sum_{i=1}^{s} p'_i (p'_i)^t R^t - 2\lambda_4 \sum_{i=1}^{s} p'_i (q'_i)^t =$$

$$2\lambda_4 \sum_{i=1}^{s} \left(p'_i (p'_i)^t\right) \begin{pmatrix} R_{11} & R_{21} & R_{31} \\ R_{12} & R_{22} & R_{32} \\ R_{13} & R_{23} & R_{33} \end{pmatrix} - 2\lambda_4 \sum_{i=1}^{s} p'_i (q'_i)^t. \tag{30}$$

Formula (30) shows that the gradient $(\nabla J_4(R))^t$ splits into three independent parts

$$2\lambda_4 \sum_{i=1}^{s} \left(p'_i (p'_i)^t\right) \begin{pmatrix} R_{11} \\ R_{12} \\ R_{13} \end{pmatrix} - 2\lambda_4 (\left(\sum_{i=1}^{s} p'_i (q'_i)^t \right)_1, \tag{31}$$

$$2\lambda_4 \sum_{i=1}^{s} \left(p'_i (p'_i)^t\right) \begin{pmatrix} R_{21} \\ R_{22} \\ R_{23} \end{pmatrix} - 2\lambda_4 (\left(\sum_{i=1}^{s} p'_i (q'_i)^t \right)_2, \tag{32}$$

$$2\lambda_4 \sum_{i=1}^{s} \left(p'_i (p'_i)^t\right) \begin{pmatrix} R_{11} \\ R_{12} \\ R_{13} \end{pmatrix} - 2\lambda_4 (\left(\sum_{i=1}^{s} p'_i (q'_i)^t \right)_3, \tag{33}$$

where $(\left(\sum_{i=1}^{s} p'_i (q'_i)^t \right)_j$ means j-th column of the matrix $\sum_{i=1}^{s} p'_i (q'_i)^t$, $j = 1, 2, 3$.

Formulas (21)–(29) and (31)–(33) show that the full gradient $(\nabla J(R))^t$ split into three independent parts with respect to variables $(R_{11}\ R_{12}\ R_{13})^t$, $(R_{21}\ R_{22}\ R_{23})^t$, and $(R_{31}\ R_{32}\ R_{33})^t$.

2.6. Solution to the Variational Problem

First, we solve the variation problem (16) without the restriction $R_* \in SO(3)$, i.e., we look for a solution in the class of affine transformations. Let us equate the gradient $(\nabla J(R))^t$ to zero. M_{123} and M_4 denote following matrices:

$$M_{123} = (\sum_{i=1}^{s} \left(r_{i1}{}^p (r_{i1}{}^p)^t \right) + \left(r_{i2}{}^p (r_{i2}{}^p)^t \right) + \left(r_{i3}{}^p (r_{i3}{}^p)^t \right)) \tag{34}$$

$$M_4 = \sum_{i=1}^{s} \left(p'_i (p'_i)^t \right) \tag{35}$$

So, three independent linear systems of equations are obtained

$$(\lambda_1 M_{123} + \lambda_4 M_4) \begin{pmatrix} R_{11} \\ R_{12} \\ R_{13} \end{pmatrix} = \lambda_1 (\sum_{i=1}^{s} \langle e_1, r_{i1}{}^q \rangle e_1 (r_{i1}{}^p)^t)^t{}_1 + \lambda_1 (\sum_{i=1}^{s} \langle e_1, r_{i2}{}^q \rangle e_1 (r_{i2}{}^p)^t)^t{}_1 +$$

$$\lambda_1 (\sum_{i=1}^{s} \langle e_1, r_{i3}{}^q \rangle e_1 (r_{i3}{}^p)^t)^t{}_1 + \lambda_4 (\left(\sum_{i=1}^{s} p'_i (q'_i)^t \right)_1, \tag{36}$$

$$(\lambda_2 M_{123} + \lambda_4 M_4) \begin{pmatrix} R_{21} \\ R_{22} \\ R_{23} \end{pmatrix} = \lambda_2 (\sum_{i=1}^{s} \langle e_2, r_{i1}{}^q \rangle e_2 (r_{i1}{}^p)^t)^t{}_2 + \lambda_2 (\sum_{i=1}^{s} \langle e_2, r_{i2}{}^q \rangle e_2 (r_{i2}{}^p)^t)^t{}_2 +$$

$$\lambda_2 (\sum_{i=1}^{s} \langle e_2, r_{i3}{}^q \rangle e_2 (r_{i3}{}^p)^t)^t{}_2 + \lambda_4 (\left(\sum_{i=1}^{s} p'_i (q'_i)^t \right)_2, \tag{37}$$

$$(\lambda_3 M_{123} + \lambda_4 M_4) \begin{pmatrix} R_{31} \\ R_{32} \\ R_{33} \end{pmatrix} = \lambda_3 (\sum_{i=1}^{s} \langle e_3, r_{i1}{}^q \rangle e_3 (r_{i1}{}^p)^t)^t{}_3 + \lambda_3 (\sum_{i=1}^{s} \langle e_3, r_{i2}{}^q \rangle e_3 (r_{i2}{}^p)^t)^t{}_3 +$$

$$\lambda_3 (\sum_{i=1}^{s} \langle e_3, r_{i3}{}^q \rangle e_3 (r_{i3}{}^p)^t)^t{}_3 + \lambda_4 (\left(\sum_{i=1}^{s} p'_i (q'_i)^t \right)_3. \tag{38}$$

Remark 1. *Formulas (36)–(38) allow parallel computation of the rows of the matrix R.*

After obtaining an affine solution R, it can be projected onto $SO(3)$. The orthogonal solution is given as

$$R_* = USV^t \tag{39}$$

where the matrices U and V are the elements of the singular value decomposition $U\mathcal{D}V^t$ of the matrix $A\prime$ (assume that $\mathcal{D}$ is a diagonal matrix with nonincreasing diagonal entries), and S is defined as

$$S = \begin{cases} I, & if \det(U)\det(V) = 1 \\ diag(1, 1, -1), & if \det(U)\det(V) = -1 \end{cases} \tag{40}$$

The translation vector can be computed as

$$T_* = \frac{1}{s} \sum_{i=1}^{s} q_i - R_* p_i \tag{41}$$

Note that in Formula (41), we take points p_i and q_i from the original point clouds P and Q without centralization.

2.7. Selection of Parameters

In [35], manual selection of the parameters of the λ-functional for a given point pair (P, Q) was used. Here we propose an automatic selection of the parameters of of the λ_r-functional parameters. Let k be the number of points in the neighborhood. Let us compute all matrices $\Lambda_{i1}{}^q$, $\Lambda_{i2}{}^q$, and $\Lambda_{i3}{}^q$. Next, we define the matrix

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \tag{42}$$

as

$$\Lambda = \frac{1}{s} \sum_{i=1}^{s} \Lambda_{i1}{}^q + \Lambda_{i2}{}^q + \Lambda_{i3}{}^q \tag{43}$$

The obtained values λ_1, λ_2, and λ_3 are used in (10). Note that this method for calculating λ_1, λ_2, and λ_3 was found experimentally.

If the values λ_1, λ_2, and λ_3 are fixed, we have two variable parameters, k and λ_4. In this paper, two neighborhood sizes are used, that is, $k = [0.15 * s]$ and $k = [0.85 * s]$. Let us apply the following values of λ_4:

$$\lambda_{4j} = 10^{-8} \cdot 4^j,\ j = 1, .., 16 \tag{44}$$

Thus, one can obtain 16·2 = 32 parameter values and corresponding 32 transformations (R_*, T_*).

Suppose that the point clouds P and Q are given. Let us consider a point p_i in P and compute (by the k-d tree algorithm) the closest point q_i in Q. If the distance between p_i and q_i is less than the value of the parameter δ, then we call p_i a suitable point. The total number of suitable points is called the largest common pointset (LCP) parameter.

In our case, 32 LCP values corresponding to geometric transformations are calculated. After that, we select the transformation corresponding to the minimum value of LCP.

3. Vector Orientation Predictor and Full Registration Algorithm

3.1. Vector Orientation Predictor

It is known that eigendecomposition algorithms produce eigenvectors with unpredictable orientation. This problem is described, in particular, in [24,41]. The performance of the ICP algorithm based on the proposed λ_r-functional critically depends on the quality of the eigenvector orientation predictor. In [35], we described a method to obtain a predictable orientation of vectors. Let us recall this algorithm and then modify it.

Let p_i and q_i be the corresponding points in the clouds P and Q, respectively. Consider the k-neighborhoods of p_i and q_i and compute the corresponding covariance matrices. The eigendecomposition of these matrices gives us the orthogonal matrices $\Re_i{}^p$ and $\Re_i{}^q$. Recall that these matrices consist of the vector-columns $(r_{i1}{}^p\ r_{i2}{}^p\ r_{i3}{}^p)$ and $(r_{i1}{}^q\ r_{i2}{}^q\ r_{i3}{}^q)$.

Suppose that eigenvalues of the covariance matrices are in ascending order. In this case, the vector $r_{i3}{}^p$ corresponds to the main axis of inertia of the neighborhood, and vector $r_{i1}{}^p$ corresponds to the normal vector, provided that the neighborhood is sufficiently flat.

The vector sets $(r_{i1}{}^p\ r_{i2}{}^p\ r_{i3}{}^p)$ and $(r_{i1}{}^q\ r_{i2}{}^q\ r_{i3}{}^q)$ to define the local vector frames in P and Q. The origins of these frames are mass centers of the neighborhoods in P and Q. Assume that the orientations of vectors of the frame in Q are fixed.

The goal is to choose such orientations for the vectors of the local frame in P that they are co-oriented with those in Q.

Consider the vector $r_{i3}{}^q$ and the center of mass $\hat{q}_i$ of the neighborhood in Q. The vector $r_{i3}{}^q$ and the point $\hat{q}_i$ define the oriented x-axis of the local frame. Let us consider the orthogonal projections of all points of the cloud Q onto the x. q_{ent} and q_{ext} denote the first and last projections on the x-axis. Consider a uniform partition of the line segment $[q_{ent}; q_{ext}]$ by points $\left\{q_{int(0)} = q_{ent}, q_{int(1)}, \ldots, q_{int(m)} = q_{ext}\right\}$, where m is the number of subsegments. Leat us consider each subsegment $\left[q_{int(j-1)}; q_{int(j)}\right]$. We denoted by $q\prime_k$ the projection, the point q_k onto the axis, if this projection belongs to the considered subsegment.

c_j denotes the center of subsegment $\left[q_{int(j-1)}; q_{int(j)}\right]$, $j = 1, \ldots, m$. Let us associate the value d_j with the subsegment as

$$d_j = \frac{1}{\sum_{q_j\prime \in [q_{int(j-1)}; q_{int(j)}]} \left(|c_j - q_{int(j)}| - |c_j - q_j\prime|\right)} \cdot \sum_{q_j\prime \in [q_{int(j-1)}; q_{int(j)}]} \left(\left|c_j - q_{int(j)}\right| - \left|c_j - q_j\prime\right|\right) \parallel q_j - q_j' \parallel_{L_2}. \tag{45}$$

$ds_{i3}{}^q$ denotes the following vector:

$$ds_{i3}{}^q = (d_1, \ldots, d_m) \tag{46}$$

We call the vector $ds_{i3}{}^q$ the descriptor of the third eigenvector of the local frame in Q. Let us compute similar descriptors $ds_{i3}{}^p(+)$ and $ds_{i3}{}^p(-)$. The vector $ds_{i3}{}^p(-)$ is the inverse version of the vector $ds_{i3}{}^p(+)$.

Compare vectors $ds_{i3}{}^q$ with $ds_{i3}{}^p(+)$ and $ds_{i3}{}^q$ with $ds_{i3}{}^p(-)$ using the L_2 norm. The smallest norm value corresponds to the orientation of the third eigenvector $r_{i3}{}^p$ in the local frame in P.

Similarly, we compute the descriptors $ds_{i1}{}^q$ in Q and $ds_{i1}{}^p$ in P and vector $r_{i1}{}^p$. The vector $r_{i2}{}^p$ is the vector product $r_{i3}{}^p \times r_{i1}{}^p$.

Remark 2. *The descriptors $(ds_{i1}{}^p, ds_{i2}{}^p, ds_{i3}{}^p)$ and $(ds_{i1}{}^q, ds_{i2}{}^q, ds_{i3}{}^q)$ are orthogonal invariants of local neighborhoods.*

Note that the algorithm described above is a simplified version of the following approach.

Suppose that we have a mass center and a main axis of the inertia ellipsoid of the point cloud. Also, assume that the surface defined by the point cloud is triangulated. Consider a set of planes perpendicular to the axis. Each plane intersects the triangulation edges at some points. Let us compute the distances between these points and their orthogonal projections onto the axis. After that, we calculate the average distance and draw the circle in the considered plane. The circle has a radius equal to the calculated average distance. The center of the circle lies on the axis. One can assign to a set of such coaxial circles an array containing the radii of the circles. If the axis orientation changes, the array is inverted. The original and inverted variants of the array determine the agreed vector orientations as follows. Suppose we have a vector and a mass center of a local vector frame in the source point cloud. The vector and the mass center define the oriented axis.

Let us consider the corresponding axis in the target point cloud. One can compute a similar array for the axis in the target point cloud. After that, the difference between the array of the target point cloud and two arrays of the source point cloud is compared by the norm. The smaller value of the norm indicates which axis orientation in the source point cloud should be chosen. To simplify the calculations, the following version of the described approach can be used; that is, the average distance (of orthogonal projections onto the axis) is calculated for points where the projections lie between two adjacent coaxial planes. This is done instead of averaging distances of points belonging to the intersection of the co-axial plane and the triangulated surface (given by the point cloud). The described simplified version is used, since the construction of a triangulated surface is a complex computational task.

Figure 1 illustrates the described approach. The point cloud shown is from the ModelNet40 database [34].

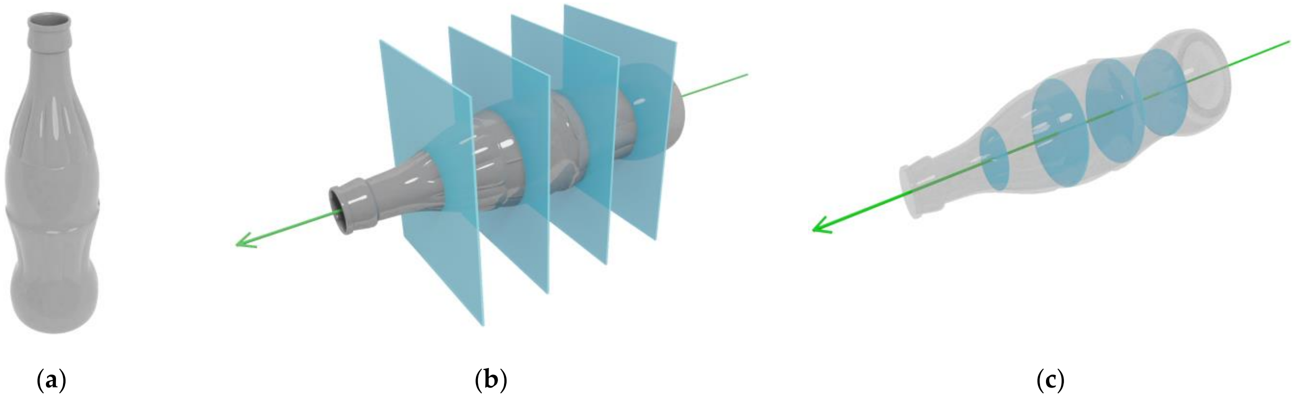


Figure 1. The illustration of the vector orientation predictor: (**a**) 3D model; (**b**) axis and planes; (**c**) the set of circles.

3.2. Full Registration Algorithm Based on λ_r-Functional

The proposed algorithm is called λ_r-ICP. In this paper, we use the λ_r-ICP algorithm for coarse registration of the incongruent point clouds.

It is known that the classical ICP algorithm cannot work well with incongruent point clouds. The ICP algorithm requires the selection of pairs of corresponding points to solve the registration problem for the incongruent case. Moreover, the common ICP algorithm is used to register point clouds for sufficiently small rotation angles. Thus, the point pair selection is based on threshold values for the distance between corresponding points.

This approach does not work in coarse registration problems because coarse registration deals with large rotation angles. The coarse registration algorithm requires the selection of pairs of points based on orthogonal invariant descriptors.

Note that the lengths of the inertia ellipsoid semi-axes are orthogonal invariants. These lengths are determined by the eigenvalues λ_{i1}, λ_{i2}, and λ_{i3}. The descriptors $({ds_{i1}}^p, {ds_{i2}}^p, {ds_{i3}}^p)$ and $({ds_{i1}}^q, {ds_{i2}}^q, {ds_{i3}}^q)$ from Section 3.1 are also orthogonal invariants.

Next, we describe the application of these orthogonal invariant descriptors in the proposed version of the λ_r-ICP algorithm. Note that the size of neighborhood k is fixed. Suppose that points p_i and q_i form a corresponding pair. Compute the local vector frames for p_i and q_i (item 2.1). Let us denote by ${\lambda_{i1}}^p$, ${\lambda_{i2}}^p$, ${\lambda_{i3}}^p$ and ${\lambda_{i1}}^q$, ${\lambda_{i2}}^q$, ${\lambda_{i3}}^q$ the eigenvalues for p_i and q_i, respectively. The parameters can be defined the follows:

$$dist\_\lambda_i = \max\{|{\lambda_{i1}}^p - {\lambda_{i1}}^q|, |{\lambda_{i2}}^p - {\lambda_{i2}}^q|, |{\lambda_{i3}}^p - {\lambda_{i3}}^q|\},\ i = 1, \ldots s \tag{47}$$

$$dist\_ds_i = max\{\| {ds_{i1}}^p - {ds_{i1}}^q \|_{L_1}, \| {ds_{i2}}^p - {ds_{i2}}^q \|_{L_1}, \| {ds_{i3}}^p - {ds_{i3}}^q \|_{L_1}\},\ i = 1, \ldots s \tag{48}$$

Let us calculate values $dist\_\lambda_i$ for all corresponding pairs and select 20% of the best pairs (i.e., with minimum values) from the total. After that, calculate values $dist\_\mathrm{ds}_i$ for all remaining pairs and select 25% of the best pairs (i.e., with minimal values). After these two selection steps, we are dealing with 5% of pairs. These steps are called the λ_r-ICP selection procedure.

Remark 3. *In the λ_r-ICP algorithm (for pairs selection), other orthogonal invariant descriptors can be used instead of those metioned above.*

Since the λ_r-ICP algorithm is used for coarse registration, we also apply the point-to-point ICP with thresholds to refine the registration result. The threshold_PtP denotes the percentage of best pairs in terms of distance between points. This parameter is used on each iteration of the point-to-point ICP to select the threshold_PtP% of pairs.

The λ_r-ICP algorithm is summarized as follows in Algorithm 1.

Algorithm 1 λ_r-ICP Algorithm

```
1. input: cloud $P$, cloud $Q$, $s$ = number of points in $P$, $K = \{[0.45 * s], [0.85 * s]\}$
2. for $counter\_1$:=1 to 2 do
  3. $k$:=$K[counter\_1]$
  4. compute local vector frames in $P$ and $Q$ for $k$-neighborhoods
  5. for $counter\_2$:=1 to 16 do
    6. $\lambda_4 = 10^{(-8)} \cdot 4^{counter\_2\text{-}1}$
    7. for $counter\_3$:= 1 to 300 do
      8. compute correspondence between $P$ and $Q$ (by k-d tree algorithm)
      9. apply vector orientation predictor for corresponding local frames in $P$ and $Q$ (here local orthogonal invariant descriptors can be changed)
      10. apply λ_r-ICP selection procedure
      11. solve λ_r-ICP variation problem and obtain geometric transformation ($R$,$T$)
      12. check the convergence criterion, if the criterion is valid then go to 14
    13. end // for $counter\_3$
    14. append obtained geometric transformation ($R$,$T$) to the list $tansform\_list$
  15. end // for $counter\_2$
16. end // for $counter\_1$
17. make refinement by point-to-point ICP algorithm with $treshold\_PtP$ to all geometric transformation from $tansform\_list$ and choose transformation ($R$,$T$) which gives the minimum LCP value for clouds $RP + T$ and $Q$
18. output: geometric transformation ($R_*$,$T_*$)
```

4. Computer Simulation

4.1. Compared Algorithms

With the help of a computer simulation, the performance the proposed algorithm λ_r-ICP, known as the Super4PCS algorithm [23], and version of the RANSAC algorithm [18] for coarse registration is illustrated and discussed.

We used the software implementation of the Super4PCS algorithm [37]. Super4PCS generates several geometric transformations and chooses one of them that is the best according to the LCP criterion. After that, refinement by the point-to-point ICP algorithm with the parameter threshold_PtP (see Section 3.2) is performed.

The λ_r-ICP algorithm is implemented as follows. This algorithm computes 32 geometric transformations and uses refinement for all of them. After that, the algorithm selects one of them, which is the best according to the LCP criterion.

The RANSAC is implemented similarly to the λ_r-ICP algorithm. The algorithm takes four random points in P and four random points in Q. The initial correspondence is given by indices. After that, the standard point-to point ICP algorithm is used. The algorithm generates some geometric transformations and uses refinement for all of them. After that, the algorithm selects one of them, which is the best according to the LCP criterion.

In our experiments, the choice of four points is optimal for RANSAC.

Another type of the coarse registration algorithm is described in [39] and its modified version in [40]. This algorithm uses the SHOT descriptor [24] and is based on the cluster approach. We refer to this algorithm as "Cluster".

The point-to-point ICP algorithm (PtP-ICP) with the selection of pairs of corresponding points and the GICP algorithm (PCL-GICP) [5] (its software implementation from the point cloud library (PCL) [38]) are also used. Note that the considered version of the PtP-ICP algorithm uses the selection of pairs of corresponding points and is more suitable for processing truncated point clouds than PCL-GICP.

4.2. Organisation of Experiments

All point clouds used in our experiments have a diameter ≈ 2. The following four point clouds were used in the experiments: Stanford Bunny; Airplane from the ModelNet40 database [34]; Scan_1; and Scan_2. The Bunny and Airplane clouds consist of 1024 points. The point clouds Scan_1 and Scan_2 were acquired by the Intel RealSense D435 depth camera. Using the depth camera, we made two maps of the depth of the room for the same camera position, but at different time. We converted depth maps to point clouds, which are similar but not identical. In particular, these point clouds are noisy independently of each other and have a different number of points. The first and second clouds consist of 73,340 and 73,081 points, respectively. We applied downsampling to these point clouds and acquired the clouds Scan_1 and Scan_2, which consist of 2631 and 2362 points, respectively. Clouds Scan_1 and Scan_2 are scaled to a diameter of $\approx$2. Figure 2 shows the Airplane, Bunny, Scan_1, and Scan_2 point clouds.

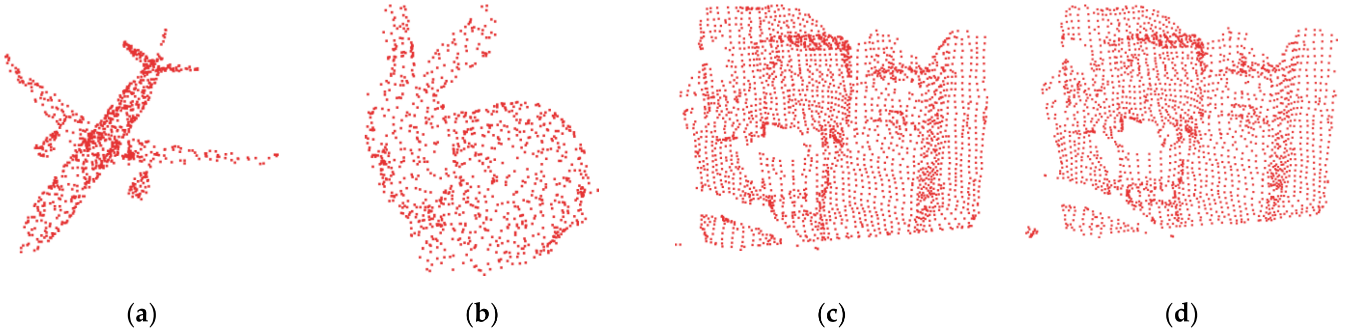


Figure 2. Initial point clouds. (**a**) Airplane; (**b**) Bunny; (**c**) Scan_1; (**d**) Scan_2.

Suppose that we have a point cloud $Cloud_{init}$. Let us fix a value of the parameter *truncate_rate* as a percentage. Calculate the mass center of the cloud and take a randomly oriented line containing the mass center. Let us project all points of $Cloud_{init}$ onto a straight line. Find the first and last projections on the line. We select truncate_rate% projections to lie on the line after the first projection. Delete corresponding points from the cloud $Cloud_{init}$. Let us denote the resulting point cloud as P_{init}.

Select truncate_rate% projections lying on the line before the last projection. Delete corresponding points from the cloud $Cloud_{init}$. Let us denote the obtained point cloud as Q_{init}.

In our experiments, the following values of the parameter *truncate_rate* are used: truncate_rate $=$ 10%, truncate_rate $=$ 20%, and truncate_rate $=$ 25%. Note that the parameter *truncate_rate* is related to the parameter overlap_rate. This parameter describes the percentage of common parts of the clouds P_{init} and Q_{init}. The corresponding values for overlap_rate are as follows: overlap_rate $\approx$ 89%, overlap_rate $=$ 75%, and overlap_rate $\approx$ 67%.

The point cloud $Q\prime$ is obtained from the point cloud Q_{init} as

$$Q' = R_{true}Q_{init} + T_{true} \tag{49}$$

Information about the rotation matrix R_{true} and the translation vector T_{true} is contained in the matrix M_{true} in homogeneous coordinates.

The point clouds P_{init} and Q' are degraded by additive Gaussian and impulse noise independently of each other. Denote the obtained point clouds as P and Q, respectively.

In the case of additive Gaussian noise, each point coordinate was corrupted by adding a Gaussian random variable with a zero mean and a standard deviation of $\sigma = 0.05$ and $\sigma = 0.10$.

All points are also distorted by impulse noise as follows: each point coordinate is changed by adding a random variable uniformly distributed in the interval $[-0.2;\ 0.2]$. The values of the noise level and truncate_rate (overlap_rate) are as follows: (σ = 0.10, truncate_rate =10% (overlap_rate ≈ 89%)); (σ = 0.05, truncate_rate = 20% (overlap_rate = 75%)); (impulse noise, truncate_rate = 10% (overlap_rate ≈ 89%)). We also carried out experiments with the Scan_1 and Scan_2 point clouds using the following conditions: only sensor noise, truncate_rate = 20% (overlap_rate = 75%). The registration task for the Airplane point cloud was also considered with conditions: noise-free, truncate_rate = 25% (overlap_rate ≈ 67%). Note that, in our experiments, we sought a balance between the level of noise and truncation, so that, on the one hand, the registration problem would be quite difficult for the compared algorithms. On the other hand, the algorithms should show a satisfactory level of results.

M_{est} denotes the matrix of geometric transformation in homogeneous coordinates, which is the result of the registration algorithm. Consider the following expression:

$$dist\_m = \| M_{true} - M_{\text{est}} \|_{L_2} \tag{50}$$

The parameter $dist\_m$ describes the difference between the true and estimated transforms. Since we are dealing with noisy point clouds, the true transformation cannot be restored exactly. Therefore, the following criterion for the quality of the result of the registration algorithm is applied. We say that the result is good if $0 \le dist\_m < 0.2$, and the result is medium if $0.2 \le dist\_m < 0.6$.

Remark 4. *The parameter dist_m evaluates the quality of the registration result for noisy incongruent point clouds with a single number. The selected parameter values work only for point clouds scaled to a diameter of about 2.*

Figure 3a shows the initial point clouds P and Q. Figure 3b shows the registration result with the parameter $dist\_m = 0.112$. Figure 3c shows the registration result with the parameter $dist\_m = 0.204$. Figure 3d shows the registration result with the parameter $dist\_m = 2.827$.

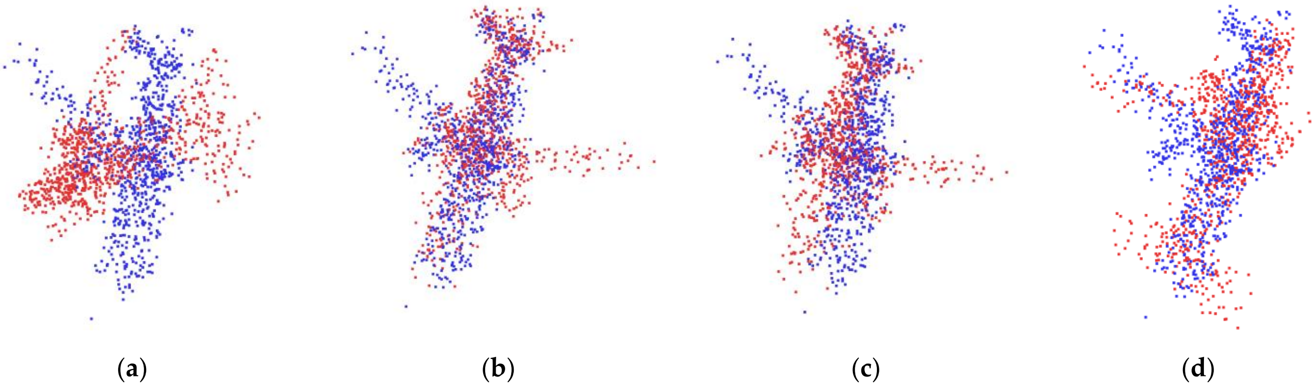


Figure 3. (**a**) Initial point clouds P and Q; (**b**) registration result with the parameter $dist\_m = 0.112$; (**c**) registration result with the parameter $dist\_m = 0.204$; and (**d**) registration result with the parameter $dist\_m = 2.827$.

The statistical experiments are organized as follows. Let us fix the value of the rotation angle. We took a random, uniformly distributed direction vector defining a line containing the origin of the coordinate system. This line is the axis of rotation at a fixed angle. In addition, the components of the translation vector are a random variable uniformly distributed in the interval $[0;1]$. The synthesized geometric transformation matrix (true matrix M_{true}) was applied to source cloud P. The algorithms were tested on the clouds P and Q. To guarantee statistically correct results, 500 trials for each fixed rotation angle were carried out. The rotation angle varied from 0 to 180 degrees with a step of 30 degrees. The number of trials satisfied to the condition of good and medium results was counted.

The percentage of good results ($0 \leq dist\_m < 0.2$) is shown in the graphs. Note that the notation "Convergence rate" for this value is also used in the graphs. The abscissa axis shows the values of rotation angles in degrees. Good and medium results ($0 \leq dist\_m < 0.6$) are also shown in the graphs.

Information about the algorithm parameters are given in the tables. The parameter "Number of ICP iterations" refers to the maximum number of ICP iterations per trial.

Our experiments were carried out on a standard desktop with Ryzen 7 1700XCPU, 32 Gb RAM, GPU NVIDIA GeForce 1080 Ti with 11 Gb GDRAM.

4.3. Experimental Results

4.3.1. Gaussian Noise with $\sigma = 0.10$, truncate_rate = 10% (overlap_rate ≈ 89%)

Figure 4 shows examples of pairs (P, Q) with Gaussian noise with $\sigma = 0.10$ and truncate_rate $=$ 10% (overlap_rate $\approx$ 89%). These pairs are the input to the coarse registration algorithms under consideration.

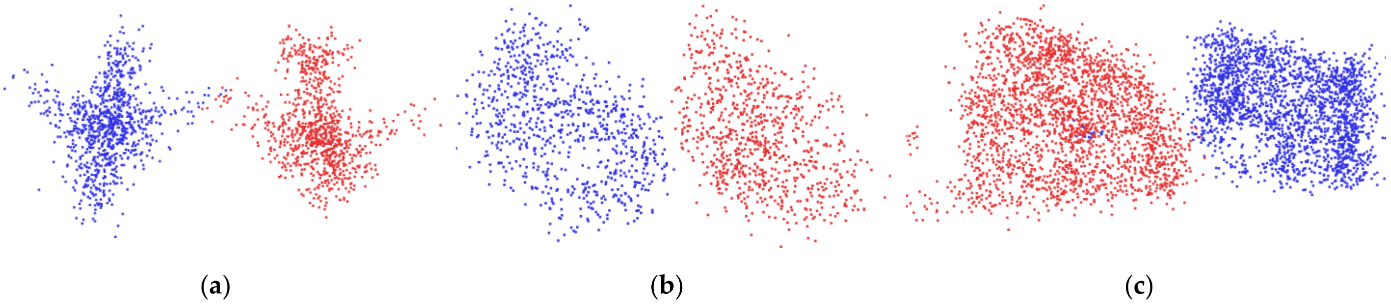


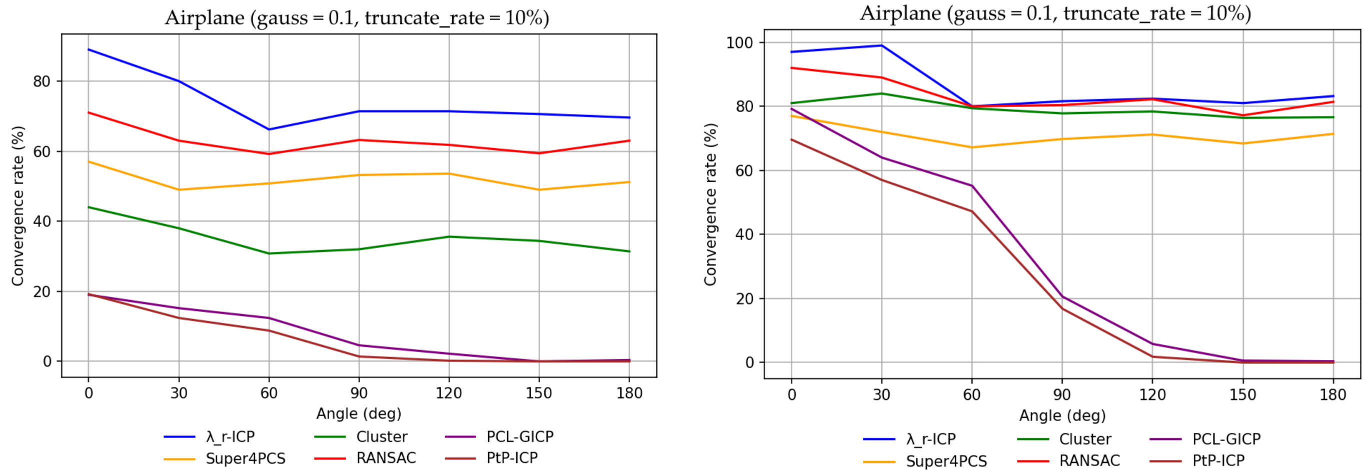
Figure 4. Examples of input data (P, Q) with Gaussian noise with $\sigma = 0.10$ and truncate_rate $=$ 10% (*o*verlap_rate $\approx$ 89%): (**a**) Airplane; (**b**) Bunny; and (**c**) Scan_1 and Scan_2.

Figure 5a shows the convergence rate of good results, i.e., the frequency of results with $0 \leq dist\_m < 0.2$ for Airplane. Figure 5b shows the convergence rate of good and medium results together, i.e., the frequency of results with $0 \leq dist\_m < 0.6$ for Airplane.

Table 1 shows the parameters of the tested algorithms for Airplane with $\sigma = 0.10$ and truncate_rate $=$ 10% (overlap_rate $\approx$ 89%).

Table 1. Parameters of the tested algorithms for Airplane with $\sigma = 0.10$ and truncate_rate $=$ 10% (overlap_rate $\approx$ 89%).

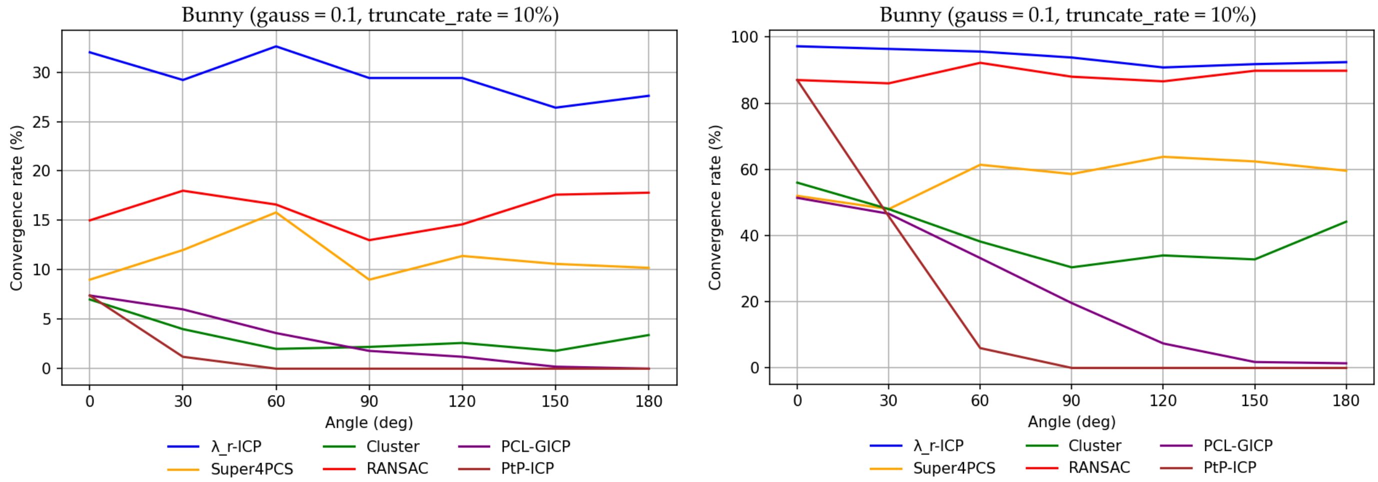
	Total Time per 3500 Trials, s	**Number of ICP Iterations**	**Threshold PtP**	**Delta (LCP)**	**Nsamples (Super4PCS)**	**Overlap (Super4PCS)**
λ_r-ICP	28,958	300	0.95	0.06	-	-
RANSAC	29,192	-	0.95	0.06	-	-
Super 4PCS	28,751	-	0.95	0.06	110	0.85
Cluster	8253	-	0.95	0.06	-	-
PCL-GICP	592	300	-	-	-	-
PtP-ICP	117	300	0.95	-	-	-



(**a**) (**b**)

Figure 5. The graphs for the Airplane point cloud, noisy with Gaussian noise with $\sigma = 0.10$ and truncate_rate $= 10\%$ (overlap_rate $\approx 89\%$): (**a**) frequency (convergence rate) of good results ($0 \leq dist\_m < 0.2$); and (**b**) frequency (convergence rate) of good and medium results ($0 \leq dist\_m < 0.6$).

Figure 6a shows the convergence rate of good results, i.e., the frequency of results with $0 \leq dist\_m < 0.2$ for Bunny. Figure 6b shows the convergence rate of good and medium results together, i.e., the frequency of results with $0 \leq dist\_m < 0.6$ for Bunny.



(**a**) (**b**)

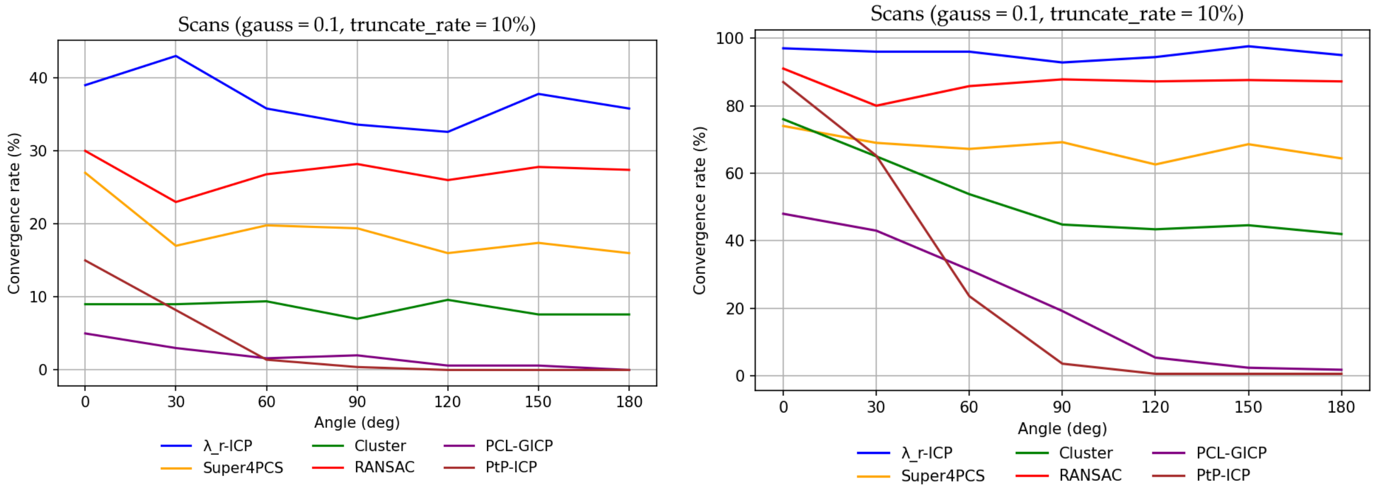
Figure 6. The graphs for Bunny point cloud, noisy with Gaussian noise with $\sigma = 0.10$ and truncate_rate $= 10\%$ (overlap_rate$\approx 89\%$): (**a**) frequency (convergence rate) of good results ($0 \leq dist\_m < 0.2$); and (**b**) frequency (convergence rate) of good and medium results ($0 \leq dist\_m < 0.6$).

Table 2 shows the parameters of the tested algorithms for Bunny, with $\sigma = 0.10$ and truncate_rate $= 10\%$ (overlap_rate $\approx 89\%$).

Figure 7a shows the convergence rate of good results, i.e., the frequency of results with $0 \leq dist\_m < 0.2$ for the Scans. Figure 7b shows the convergence rate of good and medium results together, i.e., the frequency of results with $0 \leq dist\_m < 0.6$ for the Scans.

Table 2. Parameters of the considered algorithms for Bunny with $\sigma = 0.10$ and truncate_rate $= 10\%$ (overlap_rate $\approx 89\%$).

	Total Time per 3500 Trials, s	**Number of ICP Iterations**	**Threshold PtP**	**Delta (LCP)**	**Nsamples (Super4PCS)**	**Overlap (Super4PCS)**
λ_r-ICP	32,853	300	0.95	0.06	-	-
RANSAC	29,344	-	0.95	0.06	-	-
Super 4PCS	33,665	-	0.95	0.06	120	0.85
Cluster	4909	-	0.95	0.06	-	-
PCL-GICP	371	300	-	-	-	-
PtP-ICP	114	300	0.95	-	-	-



(**a**) (**b**)

Figure 7. The graphs for Scan_1 and Scan_2 point clouds noisy by Gaussian noise with $\sigma = 0.10$ and truncate_rate $= 10\%$ (overlap_rate $\approx 89\%$): (**a**) frequency (convergence rate) of good results ($0 \leq dist\_m < 0.2$); and (**b**) frequency (convergence rate) of good and medium results ($0 \leq dist\_m < 0.6$).

Table 3 shows the parameters of the tested algorithms for Scan_1 and Scan_2 with $\sigma = 0.10$ and truncate_rate $= 10\%$ (overlap_rate $\approx 89\%$).

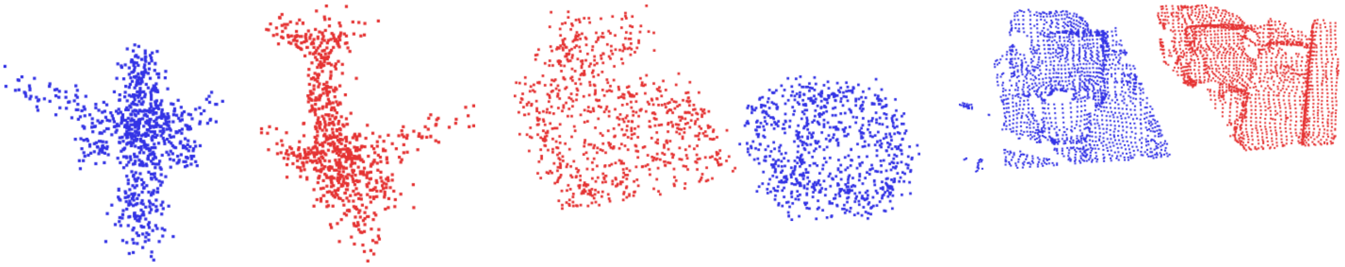
Table 3. Parameters of the tested algorithms for Scan_1 and Scan_2 with $\sigma = 0.10$ and truncate_rate $= 10\%$ (overlap_rate $\approx 89\%$).

	Total Time per 3500 Trials, sec	**Number of ICP Iterations**	**Threshold PtP**	**Delta (LCP)**	**Nsamples (Super4PCS)**	**Overlap (Super4PCS)**
λ_r-ICP	30,194	300	0.95	0.06	-	-
RANSAC	28,920	-	0.95	0.06	-	-
Super 4PCS	30,090	-	0.95	0.06	120	0.85
Cluster	5589	-	0.95	0.06	-	-
PCL-GICP	319	300	-	-	-	-
PtP-ICP	119	300	0.95	-	-	-

4.3.2. Gaussian Noise with $\sigma = 0.05$, truncate_rate $= 20\%$ (overlap_rate $= 75\%$)

Figure 8 shows examples of pairs (P, Q) with Gaussian noise with $\sigma = 0.05$ and truncate_rate = 20% (overlap_rate $\approx 75\%$). These pairs are the input to the coarse registration algorithms under consideration.

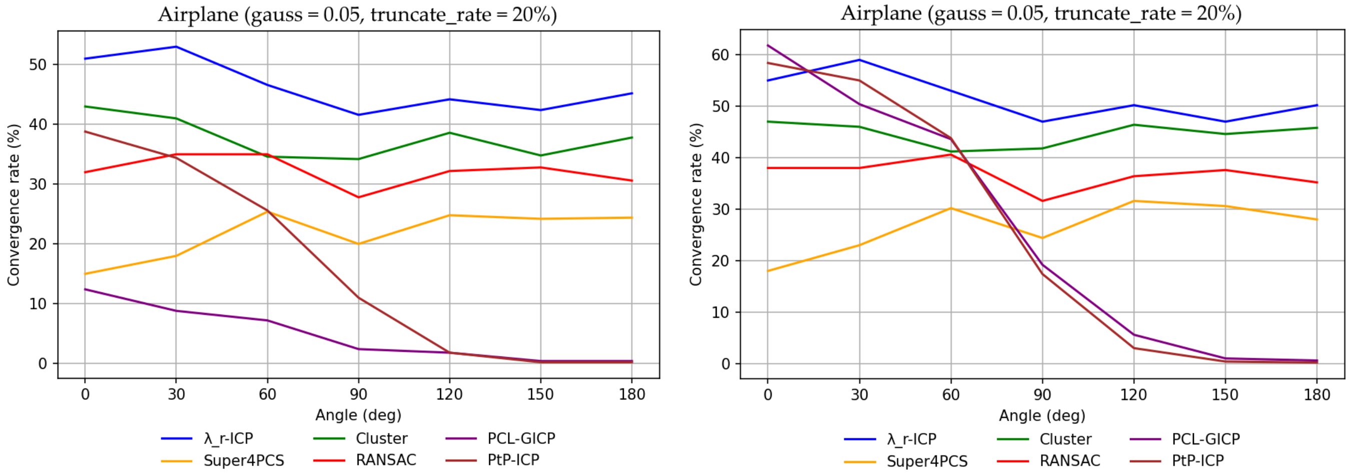
Note that we consider here point clouds Scan_1 and Scan_2 without additional noise, only with sensor noise.



(**a**) (**b**) (**c**)

Figure 8. Examples of input data (P, Q) with Gaussian noise with $\sigma = 0.05$ and truncate_rate $= 20\%$ (overlap_rate= 75%): (**a**) Airplane; (**b**) Bunny; and (**c**) Scan_1 and Scan_2.

Figure 9a shows the convergence rate of good results, i.e., the frequency of results with $0 \leq dist\_m < 0.2$ for Airplane. Figure 9b shows the convergence rate of good and medium results together, i.e., the frequency of results with $0 \leq dist\_m < 0.6$ for Airplane.



(**a**) (**b**)

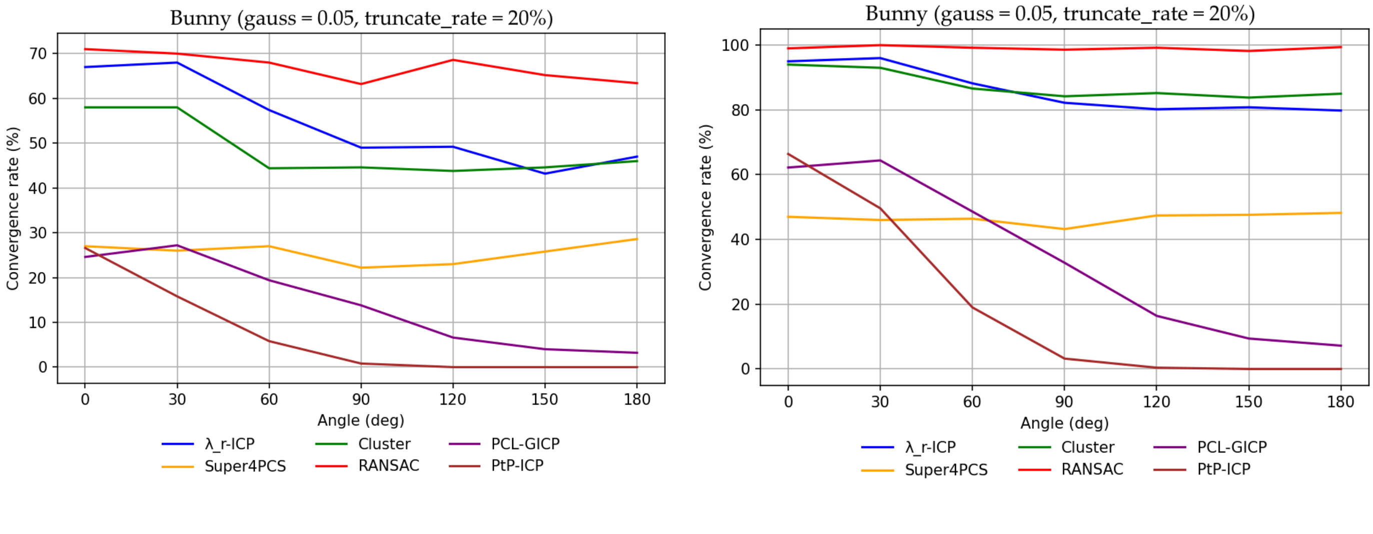
Figure 9. The graphs for Airplane point cloud noisy with Gaussian noise with $\sigma = 0.05$ and truncate_rate $= 20\%$ (overlap_rate $= 75\%$): (**a**) frequency (convergence rate) of good results ($0 \leq dist\_m < 0.2$); and (**b**) frequency (convergence rate) of good and medium results ($0 \leq dist\_m < 0.6$).

Table 4 shows the parameters of the tested algorithms for Airplane with $\sigma = 0.05$ and truncate_rate $= 20\%$ (overlap_rate $= 75\%$).

Table 4. Parameters of the tested algorithms for Airplane with $\sigma = 0.05$ and truncate_rate $= 20\%$ (overlap_rate $= 75\%$).

	Total Time per 3500 Trials, s	Number of ICP Iterations	Threshold PtP	Delta (LCP)	Nsamples (Super4PCS)	Overlap (Super4PCS)
λ_r-ICP	24,731	300	0.85	0.06	-	-
RANSAC	28,989	-	0.85	0.06	-	-
Super 4PCS	29,694	-	0.85	0.06	110	0.75
Cluster	4404	-	0.85	0.06	-	-
PCL-GICP	280	300	-	-	-	-
PtP-ICP	88	300	0.85	-	-	-

Figure 10a shows the convergence rate of good results, i.e., the frequency of results with $0 \leq dist\_m < 0.2$ for Bunny. Figure 10b shows the convergence rate of good and medium results together, i.e., the frequency of results with $0 \leq dist\_m < 0.6$ for Bunny.



(**a**) (**b**)

Figure 10. The graphs for Bunny point cloud noisy with Gaussian noise with $\sigma = 0.05$ and truncate_rate $=$ 20% (overlap_rate $=$ 75%): (**a**) frequency (convergence rate) of good results ($0 \leq dist\_m < 0.2$); (**b**) Frequency (Convergence rate) of good and medium results ($0 \leq dist\_m < 0.6$).

Table 5 shows the parameters of the tested algorithms for Bunny with $\sigma = 0.05$ and truncate_rate $=$ 20% (overlap_rate $=$ 75%).

Table 5. Parameters of the tested algorithms for Bunny with $\sigma = 0.05$ and truncate_rate $=$ 20% (overlap_rate $=$ 75%).

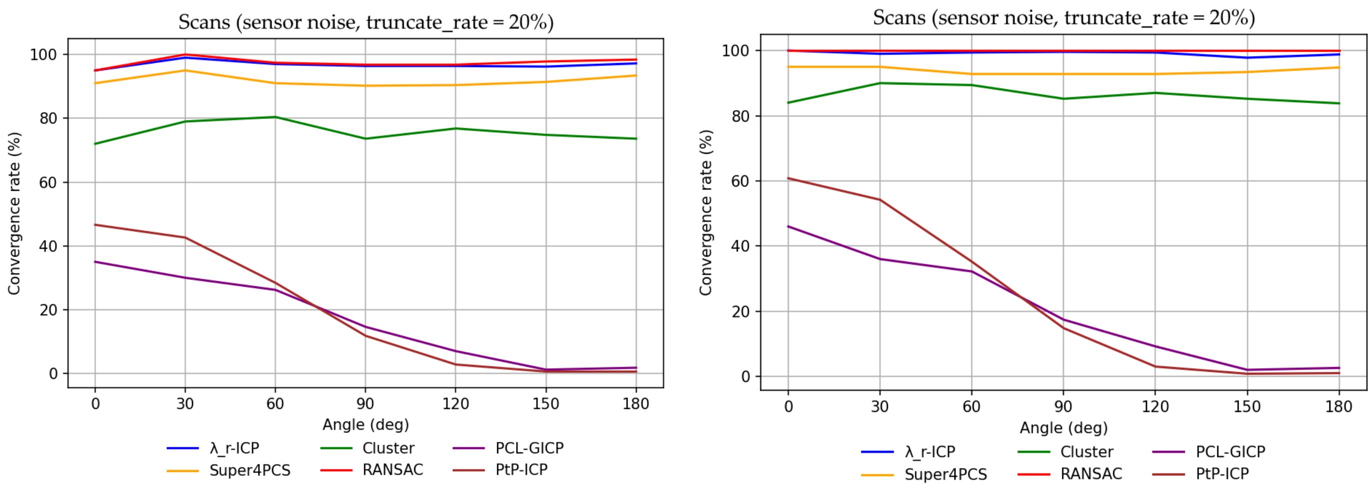
	Total Time per 3500 Trials, s	**Number of ICP Iterations**	**Threshold PtP**	**Delta (LCP)**	**Nsamples (Super4PCS)**	**Overlap (Super4PCS)**
λ_r-ICP	29,243	300	0.85	0.06	-	-
RANSAC	29,238	-	0.85	0.06	-	-
Super 4PCS	22,966	-	0.85	0.06	120	0.75
Cluster	4740	-	0.85	0.06	-	-
PCL-GICP	350	300	-	-	-	-
PtP-ICP	100	300	0.85	-	-	-

Figure 11a shows the convergence rate of good results, i.e., the frequency of results with $0 \leq dist\_m < 0.2$ for the Scans. Figure 11b shows the convergence rate of good and medium results together, i.e., the frequency of results with $0 \leq dist\_m < 0.6$ for the Scans.

Table 6 shows the parameters of the tested algorithms for Scan_1 and Scan_2 with $\sigma = 0.05$ and truncate_rate $=$ 20% (overlap_rate $=$ 75%).

Table 6. Parameters of the tested algorithms for Scan_1 and Scan_2 with $\sigma = 0.05$ and truncate_rate $=$ 20% (overlap_rate $=$ 75%).

	Total Time per 3500 Trials, s	**Number of ICP Iterations**	**Threshold PtP**	**Delta (LCP)**	**Nsamples (Super4PCS)**	**Overlap (Super4PCS)**
λ_r-ICP	27,109	300	0.85	0.04	-	-
RANSAC	29,026	-	0.85	0.04	-	-
Super 4PCS	31,291	-	0.85	0.04	120	0.75
Cluster	2113	-	0.85	0.04	-	-
PCL-GICP	301	300	-	-	-	-
PtP-ICP	77	300	0.85	-	-	-

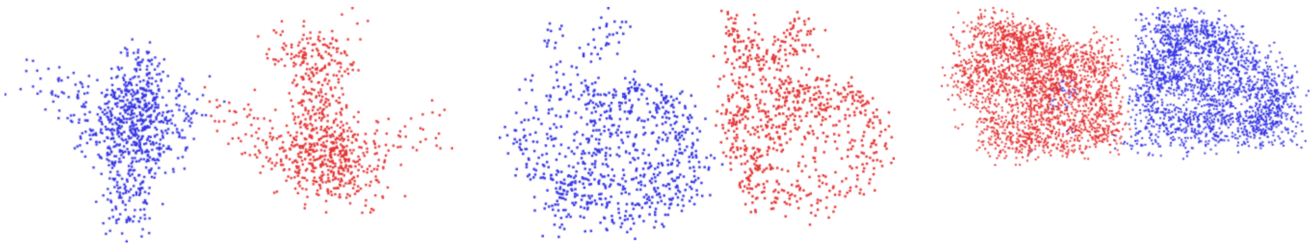


(**a**) (**b**)

Figure 11. The graphs for Scan_1 and Scan_2 point clouds noisy with Gaussian noise with the own sensor noise and truncate_rate = 20% (overlap_rate = 75%): (**a**) frequency (convergence rate) of good results ($0 \le dist\_m < 0.2$); (**b**) frequency (convergence rate) of good and medium results ($0 \le dist\_m < 0.6$).

4.3.3. Impulse Noise with, truncate_rate = 10% (overlap_rate ≈ 89%)

The parameters of impulse noise are described in Section 4.2. Figure 12 shows examples of pairs (P, Q) with impulse noise and truncate_rate = 10% (overlap_rate≈ 89%). These pairs are the input to the coarse registration algorithms under consideration.



(**a**) (**b**) (**c**)

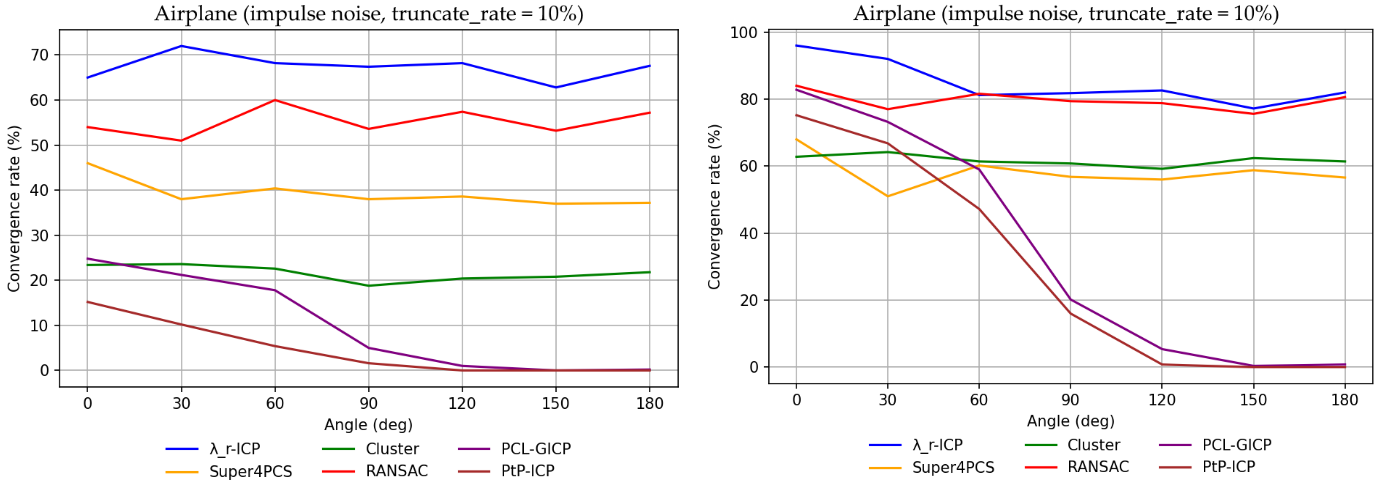
Figure 12. Examples of input data (P, Q) with impulse noise and truncate_rate = 10% (overlap_rate ≈ 89%): (**a**) Airplane; (**b**) Bunny; and (**c**) Scan_1 and Scan_2.

Figure 13a shows the convergence rate of good results, i.e., the frequency of results with $0 \le dist\_m < 0.2$ for Airplane. Figure 13b shows the convergence rate of good and medium results together, i.e., the frequency of results with $0 \le dist\_m < 0.6$ for Airplane.

Table 7 shows the parameters of the tested algorithms for Airplane with the impulse noise and truncate_rate = 10% (overlap_rate ≈ 89%).

Figure 14a shows the convergence rate of good results, i.e., the frequency of results with $0 \le dist\_m < 0.2$ for Bunny. Figure 14b shows the convergence rate of good and medium results together, i.e., the frequency of results with $0 \le dist\_m < 0.6$ for Bunny.

Table 8 shows the parameters of the tested algorithms for Bunny with the impulse noise and truncate_rate = 10% (overlap_rate ≈ 89%).

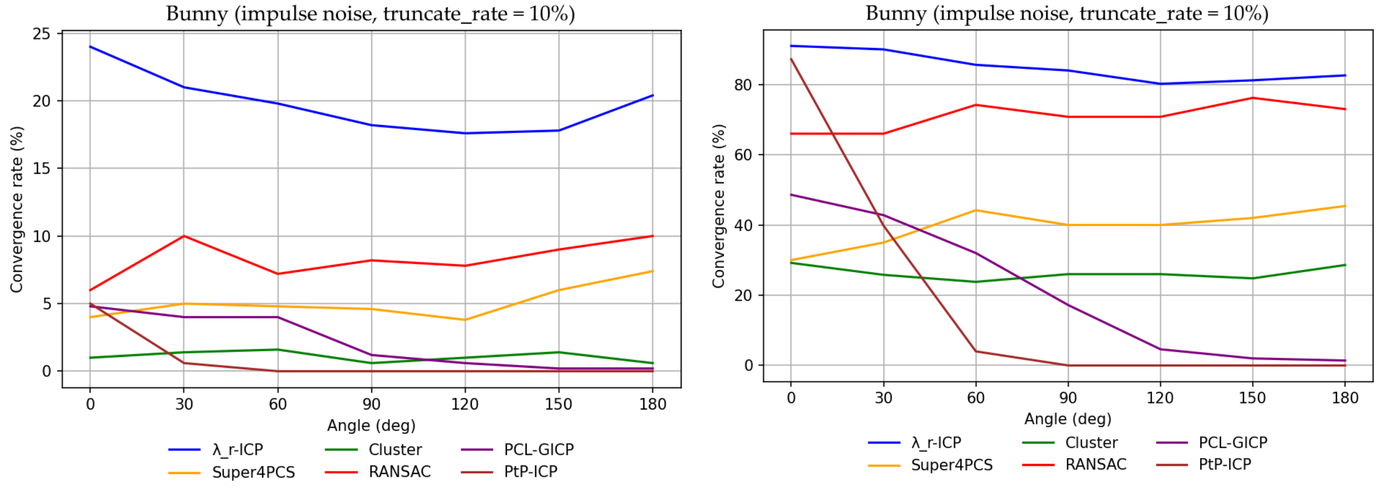


(**a**) (**b**)

Figure 13. The graphs for Airplane point cloud noisy by the impulse noise and truncate_rate = 10% (overlap_rate ≈ 89%): (**a**) frequency (convergence rate) of good results ($0 \leq dist\_m < 0.2$); (**b**) frequency (convergence rate) of good and medium results ($0 \leq dist\_m < 0.6$).

Table 7. Parameters of the tested algorithms for Airplane with the impulse noise and truncate_rate = 10% (overlap_rate ≈ 89%).

	Total Time per 3500 Trials, s	Number of ICP Iterations	Threshold PtP	Delta (LCP)	Nsamples (Super4PCS)	Overlap (Super4PCS)
λ_r-ICP	28,610	300	0.95	0.06	-	-
RANSAC	29,223	-	0.95	0.06	-	-
Super 4PCS	28,624	-	0.95	0.06	110	0.85
Cluster	6572	-	0.95	0.06	-	-
PCL-GICP	340	300	-	-	-	-
PtP-ICP	116	300	0.95	-	-	-



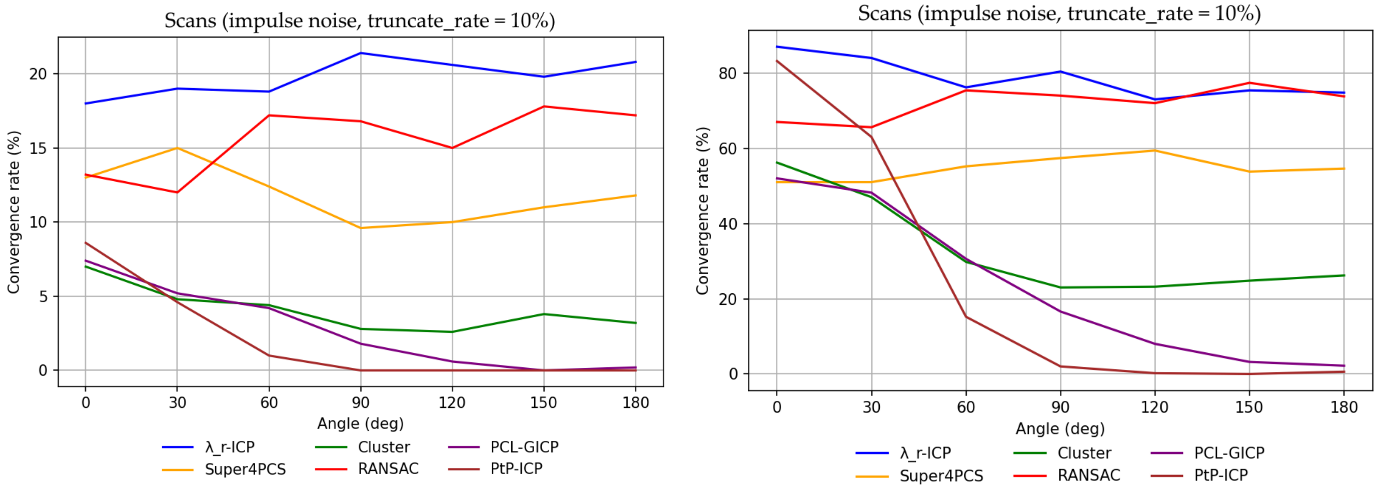
(**a**) (**b**)

Figure 14. The graphs for Bunny point cloud noisy with the impulse noise and truncate_rate = 10% (overlap_rate ≈ 89%): (**a**) frequency (convergence rate) of good results ($0 \leq dist\_m < 0.2$); and (**b**) frequency (convergence rate) of good and medium results ($0 \leq dist\_m < 0.6$).

Table 8. Parameters of the tested algorithms for Bunny with the impulse noise and truncate_rate = 10% (overlap_rate ≈ 89%).

	Total Time per 3500 Trials, s	**Number of ICP Iterations**	**Threshold PtP**	**Delta (LCP)**	**Nsamples (Super4PCS)**	**Overlap (Super4PCS)**
λ_r-ICP	29,669	300	0.95	0.06	-	-
RANSAC	29,357	-	0.95	0.06	-	-
Super 4PCS	33,560	-	0.95	0.06	120	0.85
Cluster	4599	-	0.95	0.06	-	-
PCL-GICP	373	300	-	-	-	-
PtP-ICP	108	300	0.95	-	-	-

Figure 15a shows the convergence rate of good results, i.e., the frequency of results with $0 \leq dist\_m < 0.2$ for the Scans. Figure 15b shows the convergence rate of good and medium results together, i.e., the frequency of results with $0 \leq dist\_m < 0.6$ for the Scans.



(**a**) (**b**)

Figure 15. The graphs for Scan_1 and Scan_2 point clouds noisy by the impulse noise and truncate_rate = 10% (overlap_rate≈ 89%): (**a**) Frequency (Convergence rate) of good results ($0 \leq dist\_m < 0.2$); (**b**) Frequency (Convergence rate) of good and medium results ($0 \leq dist\_m < 0.6$).

Table 9 shows the parameters of the tested algorithms for Scan_1 and Scan_2 with the impulse noise and truncate_rate = 10% (overlap_rate≈ 89%).

Table 9. Parameters of the tested algorithms for Scan_1 and Scan_2 with the impulse noise and truncate_rate = 10% (overlap_rate≈ 89%).

	Total Time per 3500 Trials, s	**Number of ICP Iterations**	**Threshold PtP**	**Delta (LCP)**	**Nsamples (Super4PCS)**	**Overlap (Super4PCS)**
λ_r-ICP	31,676	300	0.95	0.06	-	-
RANSAC	29,248	-	0.95	0.06	-	-
Super 4PCS	29,723	-	0.95	0.06	120	0.85
Cluster	5668	-	0.95	0.06	-	-
PCL-GICP	333	300	-	-	-	-
PtP-ICP	118	300	0.95	-	-	-

4.3.4. Noiseless, truncate_rate = 25% (overlap_rate ≈ 67%)

In addition, we carry out experiments with point clouds without noise and with a large truncation value (i.e., with sufficiently small common parts of clouds P and Q).

Figure 16 shows an example of point clouds P (blue color) and Q (red color) without noise and truncate_rate = 25% (overlap_rate ≈ 67%). This pair is the input to the tested coarse registration algorithms.

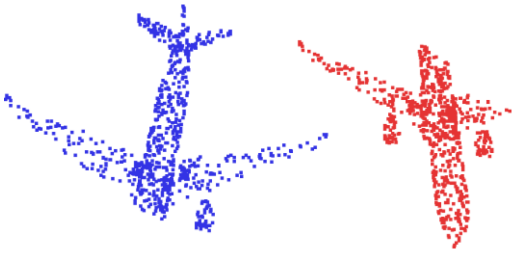
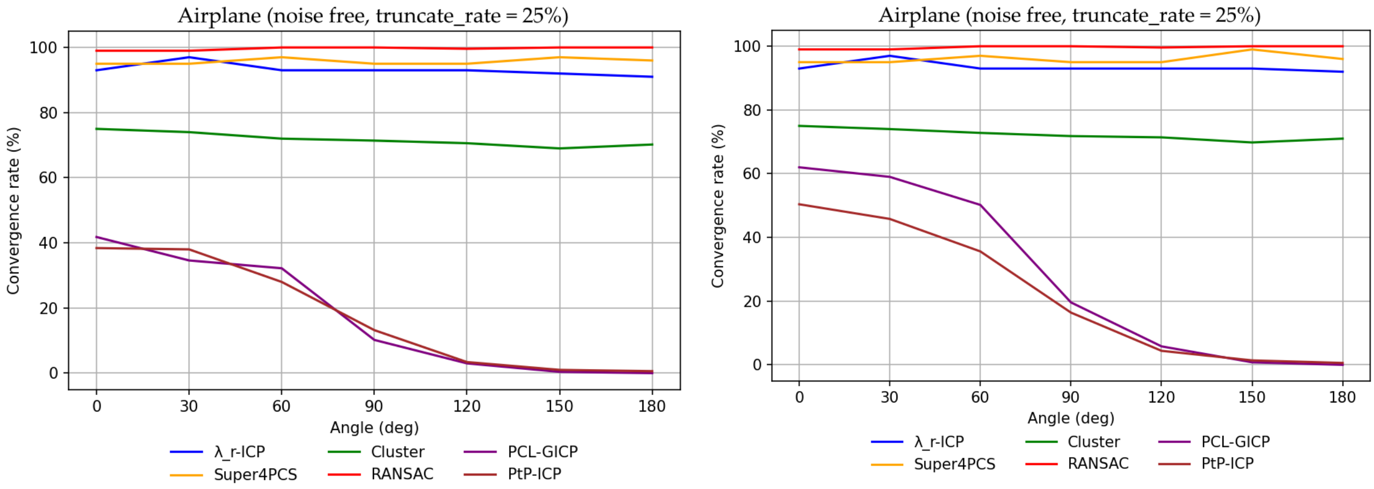


Figure 16. Example of input data P (blue color) and Q (red color) without noise and truncate_rate = 25% (overlap_rate ≈ 67%).

Figure 17a shows the convergence rate of good results, i.e., the frequency of results with $0 \leq dist\_m < 0.2$ for Airplane. Figure 17b shows the convergence rate of good and medium results together, i.e., the frequency of results with $0 \leq dist\_m < 0.6$ for Airplane.



(**a**) (**b**)

Figure 17. The graphs for airplane point cloud without noise and truncate_rate = 25% (overlap_rate ≈ 67%): (**a**) frequency (convergence rate) of good results ($0 \leq dist\_m < 0.2$); (**b**) frequency (convergence rate) of good and medium results ($0 \leq dist\_m < 0.6$).

Table 10 shows the parameters of the tested algorithms for airplane without noise and truncate_rate = 25% (overlap_rate ≈ 67%).

Table 10. Parameters of the tested algorithms for airplane without noise and truncate_rate = 25% (overlap_rate ≈ 67%).

	Total Time per 3500 Trials, s	Number of ICP Iterations	Threshold PtP	Delta (LCP)	Nsamples (Super4PCS)	Overlap (Super4PCS)
λ_r-ICP	30,097	300	0.6	0.004	-	-
RANSAC	29,302	-	0.6	0.004	-	-
Super 4PCS	5881	-	0.6	0.004	1000	0.66
Cluster	1523	-	0.6	0.004	-	-
PCL-GICP	256	300	-	-	-	-
PtP-ICP	64	300	0.6	-	-	-

5. Conclusions

In this paper, a new version of the variational problem of the ICP algorithm was considered. The introduced variational functional is a version of the λ-functional. The drawback of the previously described λ-ICP is that it is necessary to manually select the values of the parameters λ for the given source and target point clouds. Here we described a method for the automatic calculating of these parameters. An approximate solution to the corresponding variational problem using the proposed functional was presented. An efficient λ_r-ICP algorithm for the coarse registration of incongruent point clouds was also proposed. The algorithm contains elements of both the ICP and RANSAC approaches. With the help of computer simulation, the performance of the proposed algorithms was compared with that of the state-of-the-art coarse registration algorithms such as Super4PCS and RANSAC. The proposed algorithm yields the best results among tested algorithms in the case of strong noise.

Author Contributions: Conceptualization, A.M., S.V. and V.K.; methodology, A.M. and S.V.; software, A.V.; writing—original draft preparation, V.K.; writing—review and editing, V.K. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the Russian Science Foundation (grant 21-11-00095).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bellekens, B.; Spruyt, V.; Berkvens, R.; Weyn, M. A survey of rigid 3D pointcloud registration algorithms. In Proceedings of the Fourth International Conference on Ambient Computing, Applications, Services and Technologies, Rome, Italy, 24–28 August 2014; pp. 8–13.
2. Tam, G.K.; Cheng, Z.Q.; Lai, Y.K.; Langbein, F.C.; Liu, Y.; Marshall, D.; Martin, R.R.; Sun, X.F.; Rosin, P.L. Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 1199–1217. [CrossRef] [PubMed]
3. Besl, P.; McKay, N. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [CrossRef]
4. Chen, Y.; Medioni, G. Object modelling by registration of multiple range images. *Image Vis. Comput.* **1992**, *10*, 145–155. [CrossRef]
5. Segal, A.; Haehnel, D.; Thrun, S. Generalized-ICP. *Robot. Sci. Syst.* **2010**, *5*, 161–168.
6. Serafin, J.; Grisetti, G. Using extended measurements and scene merging for efficient and robust point cloud registration. *Robot. Auton. Syst.* **2017**, *92*, 91–106. [CrossRef]
7. Makovetskii, A.; Voronin, S.; Kober, V.; Voronin, A. A regularized point cloud registration approach for orthogonal transformations. *J. Glob. Optim.* **2020**, *83*, 497–519. [CrossRef]
8. Du, S.; Zheng, N.; Meng, G.; Yuan, Z. Affine Registration of Point Sets Using ICP and ICA. *IEEE Signal Process. Lett.* **2008**, *15*, 689–692.
9. Van der Sande, C.; Soudarissanane, S.; Khoshelham, K. Assessment of relative accuracy of AHN-2 laser scanning data using planar features. *Sensors* **2010**, *10*, 8198–8214. [CrossRef]
10. Makovetskii, A.; Voronin, S.; Kober, V.; Tihonkih, D. Affine registration of point clouds based on point-to-plane approach. *Procedia Eng.* **2017**, *201*, 322–330. [CrossRef]
11. Horn, B. Closed-Form Solution of Absolute Orientation Using Unit Quaternions. *J. Opt. Soc. Am. A* **1987**, *4*, 629–642. [CrossRef]
12. Horn, B.; Hilden, H.; Negahdaripour, S. Closed-form Solution of Absolute Orientation Using Orthonormal Matrices. *J. Opt. Soc. Am. Ser. A* **1988**, *5*, 1127–1135. [CrossRef]
13. Umeyama, S. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 376–380. [CrossRef]
14. Feng, Y.; Tang, J.; Su, B.; Su, Q.; Zhou, Z. Point Cloud Registration Algorithm Based on the GreyWolf Optimizer. *IEEE Access* **2020**, *8*, 143375–143382. [CrossRef]
15. Voronin, S.; Makovetskii, A.; Kober, V.; Voronin, A. The ICP algorithm based on stochastic approach. In Proceedings of the Applications of Digital Image Processing XLIV, San Diego, CA, USA, 1–5 August 2021; Volume 11842, p. 118421P.
16. Magnusson, M.; Lilienthal, A.; Duckett, T. Scan registration for autonomous mining vehicles using 3D-NDT. *J. Field Robot.* **2007**, *24*, 803–827. [CrossRef]

17. Newcombe, R.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A.; Kohli, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A. KinectFusion: Realtime dense surface mapping and tracking. In Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR), Basel, Switzerland, 26–29 October 2011.
18. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to imageanalysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
19. Irani, S.; Raghavan, P. Combinatorial and experimental results for randomized point matching algorithms. In Proceedings of the 12th Annual Symposium on Computational Geometry, Philadelphia, PA, USA, 24–26 May 1996; Volume 12, pp. 17–31.
20. Chen, C.S.; Hung, Y.P.; Cheng, J.B. RANSAC-based DARCES: A new approach to fast automatic registration ofpartially overlapping range images. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 1229–1234. [CrossRef]
21. Bazin, J.C.; Seo, Y.; Pollefeys, M. Globally optimal consensus set maximization through rotation search. In Proceedings of the Asian Conference on Computer Vision, Daejeon, Republic of Korea, 5–9 November 2012; pp. 539–551.
22. Aiger, D.; Mitra, N.J.; Cohen-Or, D. 4-points congruent sets for robust pairwise surface registration. *ACM Trans. Graph.* **2008**, *27*, 85. [CrossRef]
23. Mellado, N.; Aiger, D.; Mitra, N.J. Super 4pcs fast global pointcloud registration via smart indexing. *Comput. Graph. Forum* **2014**, *33*, 205–215. [CrossRef]
24. Salti, S.; Tombari, F.; Di Stefano, L. SHOT: Unique signatures of histograms for surface and texture description. *Comput. Vis. Image Underst.* **2014**, *125*, 251–264. [CrossRef]
25. Rusu, R.B.L.; Blodow, N.L.; Beetz, M. Fast point feature histograms (FPFH) for 3D registration. Robotics and Automation. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation ICRA '09, Kobe, Japan, 12–17 May 2009; pp. 3212–3217.
26. Le, H.M.; Do, T.T.; Hoang, T.; Cheung, N.M. SDRSAC:Semidefinite-Based Randomized Approach for Robust Point Cloud Registration without Correspondeces. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition CVPR, Long Beach, CA, USA, 16–20 June 2019; pp. 124–133.
27. Shimada, S.; Golyanik, V.; Tretschk, E.; Stricker, D.; Theobalt, C. Dispvoxnets: Non-rigid point set alignment with supervised learning proxies. In Proceedings of the International Conference on 3D Vision (3DV), Quebec City, QC, Canada, 16–19 September 2019.
28. Gojcic, Z.; Zhou, C.; Wegner, J.; Wieser, A. The perfect match: 3d point cloud matching with smoothed densities. In Proceedings of the Computer Vision and Pattern Recognition CVPR, Long Beach, CA, USA, 16–20 June 2019; pp. 5545–5554.
29. Donati, N.; Sharma, A.; Ovsjanikov, M. Deep geometric functional maps: Robust feature learning for shape correspondence. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–18 June 2020.
30. Groß, J.; Ošep, A.; Leibe, B. Alignnet-3d: Fast point cloud registration of partially observed objects. In Proceedings of the International Conference on 3D Vision(3DV), Quebec City, QC, Canada, 16–19 September 2019; pp. 623–632.
31. Lu, W.; Wan, G.; Zhou, Y.; Fu, X.; Yuan, P.; Song, S. Deepvcp: An end-to-end deep neural network for point cloud registration. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 12–21.
32. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3D classification an segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR, Honolulu, HI, USA, 21–26 July 2017; IEEE Computer Society: Boston, MA, USA, 2017; pp. 77–85.
33. Wang, Y.; Solomon, J. Deep Closest Point: Learning Representations for Point Cloud Registration. *arXiv* **2019**, arXiv:1905.03304.
34. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3D Shapenets: A Deep Representation for Volumetric Shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR, Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.
35. Makovetskii, A.; Voronin, S.; Kober, V.; Voronin, A. Point Cloud Registration Based on Multiparameter Functional. *Mathematics* **2021**, *9*, 2589. [CrossRef]
36. Makovetski, A.; Voronin, S.; Kober, V.; Voronin, A. A non-iterative method for approximation of the exact solution to the point-to-plane variational problem for orthogonal transformation. *Math. Methods Appl. Sci.* **2018**, *41*, 9218–9230. [CrossRef]
37. Super4PCS. Available online: https://github.com/nmellado/Super4PCS (accessed on 15 November 2022).
38. Point Cloud Library. Available online: https://github.com/PointCloudLibrary (accessed on 15 November 2022).
39. Makovetskii, A.; Voronin, S.; Kober, V.; Voronin, A. Registration algorithm for incongruent point clouds. In Proceedings of the 2022 VIII International Conference on Information Technology and Nanotechnology (ITNT), IEEE Proceedings, Samara, Russia, 23–27 May 2022; pp. 1–8.
40. Voronin, S.; Makovetskii, A.; Kober, V.; Voronin, A.; Turov, M. Clustering in coarse registration task and extraction of common partsnof point clouds. In Proceedings of the SPIE Applications of Digital Image Processing XLV, San Diego, CA, USA, 22–24 August 2022; Volume 12226, p. 122261F.
41. Bro, R.; Acar, E.; Kolda, T. Resolving the sign ambiguity in the singular value decomposition. *J. Chemom.* **2008**, *22*, 135–140. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.