EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH **Open Access**

Robust surface registration using *N*-points approximate congruent sets

Jian Yao*, Mauro R Ruggeri, Pierluigi Taddei and Vítor Sequeira

Abstract

Scans acquired by 3D sensors are typically represented in a local coordinate system. When multiple scans, taken from different locations, represent the same scene these must be registered to a common reference frame. We propose a fast and robust registration approach to automatically align two scans by finding two sets of *N*-points, that are approximately congruent under rigid transformation and leading to a good estimate of the transformation between their corresponding point clouds. Given two scans, our algorithm randomly searches for the best sets of congruent groups of points using a RANSAC-based approach. To successfully and reliably align two scans when there is only a small overlap, we improve the basic RANSAC random selection step by employing a weight function that approximates the probability of each pair of points in one scan to match one pair in the other. The search time to find pairs of congruent sets of *N*-points is greatly reduced by employing a fast search codebook based on both binary and multi-dimensional lookup tables. Moreover, we introduce a novel indicator of the overlapping region quality which is used to verify the estimated rigid transformation and to improve the alignment robustness. Our framework is general enough to incorporate and efficiently combine different point descriptors derived from geometric and texture-based feature points or scene geometrical characteristics. We also present a method to improve the matching effectiveness of texture feature descriptors by extracting them from an atlas of rectified images recovered from the scan reflectance image. Our algorithm is robust with respect to different sampling densities and also resilient to noise and outliers. We demonstrate its robustness and efficiency on several challenging scan datasets with varying degree of noise, outliers, extent of overlap, acquired from indoor and outdoor scenarios.

1 Introduction

In the past decade, there was a growing interest in 3-D reconstruction and realistic 3-D modelling of large-scale scenes such as urban structures. Applications of such models include virtual reality, cultural heritage, urban planning, and architecture. Commonly, these applications require a combination of laser sensing technology with traditional digital photography.

Applications that employ only digital images extract 3-D information using either a single moving camera or a multi camera system, such as a stereo rig. In both cases, the system extracts and matches distinctive features (typically points) among the available images and estimates both their 3-D positions and the camera parameters [1,2]. It is then possible to exploit the result of this first step to perform a dense point reconstruction by estimating a depth map for each image [3,4]. On one hand, these approaches are useful for those applications requiring a robust and low-cost acquisition system. On the other hand, laser sensing technology yields much higher precision and resolution. Thus, the laser sensing technology represents an effective and powerful tool for achieving accurate geometric representations of complex surfaces of real scenes.

In recent years, 3-D laser scanners able to provide satisfying measurement accuracy for different applications become commercially available. These sensors are used to acquire a complex real scene through multiple scans taken from different positions to fully describe the scene while reducing the number of occluded surfaces. For this reason, it is important to employ a systematic and automatic way to align, or register, multiple 3-D scans to represent and visualize them in a common coordinate system. Geometrically, given a point cloud $\mathcal{Q}$ considered as reference and a second point cloud $\mathcal{P}$, the

* Correspondence: jian.yao@jrc.ec.europa.eu
Institute for Transuranium Elements, European Commission-Joint Research Centre (JRC), Ispra, VA, Italy

problem of registration consists in finding the rigid transformation $\mathcal{T}$, which optimally aligns $\mathcal{P}$ to $\mathcal{Q}$ in its coordinate system.

1.1 Related works

The iterative closest point (ICP) algorithm [5] is the de facto standard to compute the rigid transformation $\mathcal{T}$ between two point clouds. It is basically an optimization method that starts from an initial estimate of $\mathcal{T}$ and iteratively refines it by generating pairs of corresponding points on the scans and minimizing an error metric, e.g., the sum of squared distances between corresponding points. Although several variants of ICP were presented [6] to improve its efficiency, the main problem is to achieve a good initial estimate of $\mathcal{T}$ since the ICP optimization can easily stop in local minima.

The problem of automatically registering two scans was achieved with a wide variety of methods [7]. Most of these extract sets of feature points, which are automatically matched to recover a good approximation of $\mathcal{T}$. Aiger et al. [8] proposed to automatically match congruent sets of four roughly coplanar points to solve the largest common point (LCP) set problem. Congruent sets of points have similar shapes defined in terms of point distances and normal deviations. The best match between congruent sets is randomly found by following the RANdom SAmple Consensus (RANSAC) approach [9]. Other approaches use shape descriptors to identify sets of candidate feature points to be matched. Gendalf et al. [10] use a 3-D integral invariant shape descriptor to detect feature points, which are matched in sets of three items using a branch-and-bound algorithm. Other interesting shape descriptors invariant with respect to rigid transformation are used to identify feature points, such as scale invariant feature transforms (SIFT)s [11,12] or Harris corners [12] extracted from reflectance images, 3-D SIFT-like descriptors extracted from triangle meshes approximating the point clouds [13], wavelet features [14], intensity-based range features [15], spin images [16,17], and extended Gaussian images [18].

Methods to automatically recover the rigid transformations from matching sets of higher-level features were also presented. The advantage of these approaches is the reduction of the search space identified by two small sets of features, which results in efficient matching, but that should account for extra computation time due to scene segmentation or feature detection. Among the feature types presented the most interesting are: lines [19,20], planes [19-22], circles [23], spheres [24] and other fitted geometric primitives [25].

Other studies proposed to formulate the registration as an energy optimization problem that does not need any explicit set of point correspondences. Silva et al. [26] proposed to use an enhanced genetic algorithm to solve the range image registration problem using a robust surface interpenetration measure. Boughorbel et al. [27] defined an energy minimization function based on Gaussian fields to solve the 3-D automatic registration.

The last relevant class of registration approaches is based on modelling the alignment of two point sets as an assignment problem, where the probability of a point in one set to has a correspondence in the other set is estimated and maximized with expectation maximization (EM) algorithms. Popular methods following this approach are known as SoftAssign [28] and EM-ICP [29], which are both based on entropy maximization principles, but imposing different constraints for problem optimization, i.e., a two-way constraint embedded into the deterministic annealing scheme for SoftAssign and a one-way constraint for the EM-ICP. A detailed review and analysis of these methods was provided in [30], where Liu proposed a method to overcome SoftAssign and EM-ICP limitations based on modelling the registration problem as Markov chain of thermodynamic systems and on an entropy model derived from the Lyapunov function. Furthermore, fast implementations on GPU of the SoftAssign and EM-ICP algorithms were recently presented by Tamaki et al. [31].

1.2 Our algorithm

Our method utilizes 3-D points (possibly associated with point descriptors, as it is described in Section 6) to achieve automated registration. It automatically aligns two scans by finding two N-points approximate congruent sets leading to a good estimate of the transformation $\mathcal{T}$ between the corresponding point clouds. $\mathcal{T}$ is then further refined via the ICP algorithm.

Given two scans $\mathcal{P}$ and $\mathcal{Q}$, our algorithm randomly searches for sets of congruent groups of points in $\mathcal{P}$ and $\mathcal{Q}$. Corresponding groups are then used to estimate a rigid transformation $\mathcal{T}$ to align $\mathcal{P}$ to $\mathcal{Q}$. The optimal transformation is recovered following a RANSAC optimization [9], which iterates the following steps until a good solution to the problem is found or the number of iterations exceeds a predefined threshold $I_{\max}$:

1) **Random selection** of a N-points base $\mathcal{B}_p$ in $\mathcal{P}$.
2) **Approximate congruent group selection** of N-points bases in $\mathcal{Q}$. The definition of approximate point set congruence is described in Section 2. This selection is achieved by using a general codebook to efficiently find approximate congruent points bases under rigid transform by exploiting combinations of feature point descriptors when available (see Section 3).
3) **Estimation** of the transformation $\mathcal{T}$ between $\mathcal{P}$ and $\mathcal{Q}$ given a randomly selected N-points base $\mathcal{B}_p$ in

$\mathcal{P}$ and each extracted approximate congruent N-points base $\mathcal{B}_p$ in $\mathcal{Q}$.

4) **Verification** of the transformation. $\mathcal{T}$ is verified using all possibly corresponding points after the alignment. The verification employs our proposed quality-based largest common pointset (QLCP) measure described in Section 4.4.

The best transformation is selected as the one yielding the best QLCP measure and then further refined with the ICP algorithm. As in [20], we present a variant of this RANSAC-based algorithm, which improves the random selection step by employing a weight function approximating the probability of each pair of features in $\mathcal{P}$ to be matched with one in $\mathcal{Q}$. We call this variant as *probability-based RANSAC approach* and describe it in Section 4.1.

Our algorithm is robust with respect to different sampling densities and the typical noise introduced by laser scanner acquisition. This is achieved by employing suitable point sampling approaches described in Section 5, and by using feature points and their descriptors to effectively constrain rigid transformations on noisy point sets.

Through our proposed matching framework presented in Sections 2 and 3, we efficiently match points and point pairs in a multi-dimensional space defined by a set of available geometric and texture feature descriptors and geometrical constrains of a set of sampled points. The matching is performed by combining suitable metric functions to compare the provided descriptors.

Any type of features carrying suitable distance functions to be compared can be easily integrated into our matching framework. The major benefit of this approach is to make possible efficient customizations for specific applications aiming at relevantly improving the registration performance in terms of robustness, accuracy, and execution time.

We also present a method to improve the matching effectiveness of texture features extracted from typical spherical reflectance images acquired by laser scanners. It consists in extracting features from atlases of rectified perspective images constructed by sampling the reflectance image spherical field of view at suitable angles. This approach mitigates the effect of spherical distortion on the resulting feature signatures so that they can be matched with higher reliability.

The robustness, accuracy and efficiency of our method were overall evaluated on several challenging scan datasets acquired from indoor and outdoor scenes as described in Section 7.

2 Approximate point set congruence

Given two point sets $\mathcal{P}$ and $\mathcal{Q}$, we assume $\mathcal{B}_p = \{\mathbf{p}_i | \mathbf{p}_i \in \mathcal{P}\}_{i=1}^N$ and $\mathcal{B}_q = \{\mathbf{q}_i | \mathbf{q}_i \in \mathcal{Q}\}_{i=1}^N$ to be the two corresponding N-points bases from $\mathcal{P}$ and $\mathcal{Q}$, respectively. This means that for each point $\mathbf{p}_i \in \mathcal{B}_p$ there exists one and only one corresponding point $\mathbf{q}_i \in \mathcal{B}_q$. We consider the two sets to be congruent, if they are approximatively similar in shape and have a similar distribution in 3-D space. We define both a similarity score function and a binary similarity score function in order to measure the congruency of two matching N-points bases as follows.

Given a point $\mathbf{p}_i \in \mathcal{B}_p$, we characterize it using a set of L local descriptors, i.e., $\{f_l(\mathbf{p}_i)\}_{l=1}^L$. Similarly, for each pair of points $(\mathbf{p}_i, \mathbf{p}_j)$ in $\mathcal{B}_p$, we define a set of K measurements $\{m_k(\mathbf{p}_i, \mathbf{p}_j)\}_{k=1}^K$. The L descriptors and the K measurements characterize a N-points base in terms of point features and point pairs relations (see Figure 1). These values are then used to define the congruency of the two different N-points bases $\mathcal{B}_p$ and $\mathcal{B}_q$.

For each type of local descriptor or points pair measurement, we define a *similarity difference* function $d(\cdot, \cdot)$ that is invariant under rigid transformation of each single N-points base. In particular, given two descriptors or measurements $\mathbf{v}_p$ and $\mathbf{v}_q$, then $d(\mathbf{v}_p, \mathbf{v}_q)$ is represented by a real positive value that states how different the two descriptors or measurements are.

We also define a set of *boolean similarity measures*

$$s(\mathbf{v}_p, \mathbf{v}_q) = b(d(\mathbf{v}_p, \mathbf{v}_q), t), \tag{1}$$

where t is a threshold value associated to the particular feature descriptor or measurement and $b(\cdot, \cdot)$ is a boolean function defined as:

$$b(x, t) = \begin{cases} 1 & x \leq t, \\ 0 & x > t. \end{cases} \tag{2}$$

The set of functions $\{d(\cdot, \cdot)\}$ are then composed together to define a *similarity score* $s_c(\mathcal{B}_p, \mathcal{B}_q)$between two congruent N-points bases as follows:

$$s_c(\mathcal{B}_p, \mathcal{B}_q) = s_c^f(\mathcal{B}_p, \mathcal{B}_q) + s_c^m(\mathcal{B}_p, \mathcal{B}_q), \tag{3}$$

where s_c^f is the term related to the local descriptors and s_c^m is the term related to the similarity measures of points pairs. These two terms are defined as:

$$s_c^f(\mathcal{B}_p, \mathcal{B}_q) = \frac{1}{N_f} \sum_{i=1}^{N} \sum_{l=1}^{L} w_l^f \cdot \left(1 - \frac{\min\left(d(f_l(\mathbf{p}_i), f_l(\mathbf{q}_\mathrm{i})), t_l^f\right)}{t_l^f}\right), \tag{4}$$

$$s_c^m(\mathcal{B}_p, \mathcal{B}_q) = \frac{1}{N_m} \sum_{\substack{i,j = 1 \cdots N \\ i \neq j}} \sum_{k=1}^{K} w_k^m \cdot \left(1 - \frac{\min(d(m_k(\mathbf{p}_i, \mathbf{p}_j), m_k(\mathbf{q}_i, \mathbf{q}_j)), t_k^m)}{t_k^m}\right), \tag{5}$$

where $\{w_l^f\}$ and $\{w_k^m\}$ are user-defined weights, $N_m = 2N(N-1)\sum_{k=1}^K w_k^m$ and

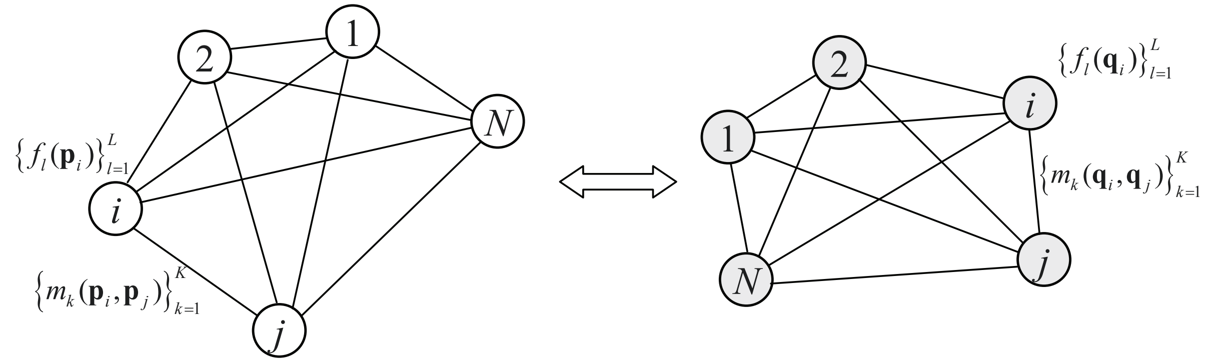

2
1
N
$\{f_l(\mathbf{p}_i)\}_{l=1}^{L}$
i
$\{m_k(\mathbf{p}_i,\mathbf{p}_j)\}_{k=1}^{K}$
j
2
$\{f_l(\mathbf{q}_i)\}_{l=1}^{L}$
i
1
$\{m_k(\mathbf{q}_i,\mathbf{q}_j)\}_{k=1}^{K}$
N
j

Figure 1 An example of two congruent *N*-points bases (*left* from $\mathcal{P}$ and *right* from $\mathcal{Q}$). For each point, we evaluate the set of *L* local features and for each points pair, we evaluate the set of *K* measurements. Features and measurements between the two sets are compared using the corresponding similarity differences.

$N_m = 2N(N-1)\sum_{k=1}^{K} w_k^m$ are normalization factors. Notice that s_c is defined such that its values fall in the range [0, 1], where a higher value represents a higher similarity between the two *N*-points bases.

Similarly, we define the *binary similarity score* $s_c(\mathcal{B}_p, \mathcal{B}_q)$ between two *N*-points bases as:

$$s(\mathcal{B}_p, \mathcal{B}_q) = \left(\prod_{i=1}^{N}\prod_{l=1}^{L} s_l(f_l(\mathbf{p}_i), f_l(\mathbf{q}_i))\right) \cdot \left(\prod_{\substack{i,j=1\cdots N\\ i\neq j}}\prod_{k=1}^{K} s_k(m_k(\mathbf{p}_i, \mathbf{p}_j), m_k(\mathbf{q}_i, \mathbf{q}_j))\right). \qquad (6)$$

where $s_c(\mathcal{B}_p, \mathcal{B}_q)$ represents the product of all boolean similarities associated to the matching points of the two sets. We consider $\mathcal{B}_p$ and $\mathcal{B}_q$ to be approximate congruent only if $s(\mathcal{B}_p, \mathcal{B}_q) = 1$.

In order to evaluate the *N*-points base congruence, we need to define which local point descriptors $\{f_l(\cdot)\}$ and points pair measurements functions $\{m_k(\cdot, \cdot)\}$ to employ, their corresponding similarity differences $\{d(\cdot, \cdot)\}$ and scalar thresholds $\{t\}$. We consider as first points pair measurement the Euclidean distance $m_1(\mathbf{p}_i, \mathbf{p}_j) = ||\mathbf{p}_i - \mathbf{p}_j||$ and define its similarity difference as:

$$d_1(m_1(\mathbf{p}_i, \mathbf{p}_j), m_1(\mathbf{q}_i, \mathbf{q}_j)) = 1 - \frac{\min(m_1(\mathbf{p}_i, \mathbf{p}_j), m_1(\mathbf{q}_i, \mathbf{q}_j))}{\max(m_1(\mathbf{p}_i, \mathbf{p}_j), m_1(\mathbf{q}_i, \mathbf{q}_j))}. \qquad (7)$$

If the surface normal at each point is available, we define the second points pair measurement $m_2(\mathbf{p}_i, \mathbf{p}_j) = n_{\text{angle}}(\mathbf{n}(\mathbf{p}_i), \mathbf{n}(\mathbf{p}_j))$, where $\mathbf{n}(\mathbf{p})$ denotes the surface normal of the point $\mathbf{p}$ and $n_{\text{angle}}(\cdot, \cdot)$ denotes the minimal angle between the two surface normals. Its similarity difference is defined as:

$$d_2(m_2(\mathbf{p}_i, \mathbf{p}_j), m_2(\mathbf{q}_i, \mathbf{q}_j)) = |m_2(\mathbf{p}_i, \mathbf{p}_j) - m_2(\mathbf{q}_i, \mathbf{q}_j)| \qquad (8)$$

If the reflectance or colour images associated with the range scans are available we can extract the corresponding feature points (e.g., SIFT or SURF feature points [32]) associated with each 3-D point of $\mathcal{B}_p$ and $\mathcal{B}_q$. The corresponding local feature descriptors can be used to define a suitable similarity difference.

In some application, it is possible to exploit some information about the environment to define additional descriptors. This is the case of scans representing structural scenes with one common and main normal direction (*ground floor scene*) or environment with three common orthogonal normal directions (*orthogonal scene*). For instance, in an indoor/outdoor scene with a common and main ground floor plane, all points lying on the ground plane roughly have the same normal directions. The type of a structural scene can be automatically detected and classified by clustering the surface normals.

In the case of *ground floor scene*, we initially transform all points in $\mathcal{P}$ and $\mathcal{Q}$ to align their corresponding ground floor normals to the *z*-axis. Then for each point $\mathbf{p} = (\mathbf{p}^x, \mathbf{p}^y, \mathbf{p}^z)^\top$, an additional local descriptor can be defined as $f_z(\mathbf{p}) = n_{\text{angle}}(\mathbf{p}, \mathbf{n}_z)$, where $\mathbf{n}_z$ denotes the direction of *z*-axis, i.e., $\mathbf{n}_z = (0, 0, 1)^\top$. $f_z(\mathbf{p})$ represents the inclination of the surface passing through $\mathbf{p}$ w.r.t. the ground. Its similarity difference is defined as:

$$d_{fz}(f_z(\mathbf{p}_i), f_z(\mathbf{q}_i)) = |f_z(\mathbf{p}_i) - f_z(\mathbf{q}_i)|. \qquad (9)$$

In addition, we can introduce another points pair measurement $m_z(\mathbf{p}_i, \mathbf{p}_j) = \mathbf{p}_i^z - \mathbf{p}_j^z$, i.e., the height difference between the two points. Its corresponding similarity difference can be defined as:

$$d_{mz}(m_z(\mathbf{p}_i, \mathbf{p}_j), m_z(\mathbf{q}_i, \mathbf{q}_j)) = |m_z(\mathbf{p}_i, \mathbf{p}_j) - m_z(\mathbf{q}_i, \mathbf{q}_j)|. \qquad (10)$$

If both $\mathcal{P}$ and $\mathcal{Q}$ are acquired from an orthogonal scene, $\mathcal{P}$ and $\mathcal{Q}$ are first transformed to align their corresponding three orthogonal point normals to the *x*-, *y*- and *z*-axis, respectively. Then, we exploit two more local descriptors defined as $f_x(\mathbf{p}) = n_{\text{angle}}(\mathbf{p}, \mathbf{n}_x)$ and $f_y(\mathbf{p}) = n_{\text{angle}}(\mathbf{p}, \mathbf{n}_y)$, where $\mathbf{n}_x = (1, 0, 0)^\top$ and $\mathbf{n}_y = (0, 1, 0)^\top$.

These descriptors represent the inclinations of the surface passing through $\mathbf{p}$ w.r.t. the additional main axes $\mathbf{n}_x$ and $\mathbf{n}_y$, respectively. In addition, we introduce two other points pair measurements $m_x(\mathbf{p}_i, \mathbf{p}_j) = \mathbf{p}_i^x - \mathbf{p}_j^x$, and $m_y(\mathbf{p}_i, \mathbf{p}_j) = \mathbf{p}_i^y - \mathbf{p}_j^y$. The corresponding similarity difference of f_x, f_y, m_x and m_y are defined in the same way as in Equations (9) and (10), respectively.

3 Fast search codebook

Using the criteria defined in the previous section, we are able to evaluate the congruency of two given N-points bases. To perform the registration, we need to couple a N-points base $\mathcal{B}_p \in \mathcal{P}$ with the N-points base $\mathcal{B}_q \in \mathcal{Q}$ having a high-similarity score. This task requires a search over all possibly congruent N-points bases in $\mathcal{Q}$. Using exhaustive search approaches is impractical due to the large number of candidates in $\mathcal{Q}$. To solve this problem, we build a codebook from $\mathcal{P}$ and $\mathcal{Q}$ composed of two different data structures used to perform a fast search of possibly corresponding points (as described in Section 3.1) and point pairs (as described in Section 3.2). In particular, we employ a boolean table S_f used to detect candidate point matches in $\mathcal{Q}$ of a selected point $\mathbf{p}_i \in \mathcal{P}$ and a multi-dimensional table S_m used to detect candidate point pair matches of a selected pair of points $(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{P}$. If the number of all detected congruent N-points bases is still large, we further need to compute a similarity score between $\mathcal{B}_p$ and each detected base in $\mathcal{Q}$ in order to sort them and then consider only the best ones.

The used codebook is, thus, composed of a boolean $m \times n$ table and a floating-point $n \times n \times K$ table[a], where $m = |\mathcal{P}|$, $n = |\mathcal{Q}|$ and K denote the number of used points pair measurements. The required memory for S_f increases as $O(mn)$ and for S_m increases as $O(n^2)$ (assuming $K \ll n$).

Our algorithm detects candidate congruent N-points bases incrementally. Given $\mathcal{B}_p \in \mathcal{P}$, we start by selecting two points in $\mathcal{B}_p$ and collect all congruent 2-points bases in $\mathcal{B}_q$. We then iteratively add points to the current selection and grow the set of candidate bases until we reach a set of N-points bases.

3.1 Point features lookup table

We build S_f as a $m \times n$ boolean similarity measure table according to the used local descriptors, where $m = |\mathcal{P}|$ and $n = |\mathcal{Q}|$, i.e., the sizes of $\mathcal{P}$ and $\mathcal{Q}$, respectively. Each element in S_f is defined as:

$$S_f(\mathbf{p}_i, \mathbf{q}_j) = \prod_{l=1}^{L} s_l(f_l(\mathbf{p}_i), f_l(\mathbf{q}_j)), \tag{11}$$

where $\mathbf{p}_i \in \mathcal{P}$ and $\mathbf{q}_j \in \mathcal{Q}$, $i = 1 \ldots m$, and $j = 1 \ldots n$. Thus, given a point $\mathbf{p}_i \in \mathcal{P}$, we can recover all possibly matching points in $\mathcal{Q}$ considering the ith row of S_f. The set of candidate matches in $\mathcal{Q}$ for $\mathbf{p}_i$ is represented by:

$$\mathcal{M}_f(\mathbf{p}_i) = \{\mathbf{q}_j | S_f(\mathbf{p}_i, \mathbf{q}_j) = 1\}_{\mathbf{q}_j \in Q}. \tag{12}$$

Notice that, to build S_f, we only make use of the local feature descriptors. Its size depends on the number of points of both point clouds. In Section 5, we describe several techniques to sample the input acquisitions in order to reduce their sizes.

3.2 Point pairs lookup table

Given a point pair $(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{P}$, we need to efficiently find candidate matching pairs in $\mathcal{Q}$. Using a blind exhaustive search, this would require a comparison with $\frac{n(n-1)}{2}$ point pairs. To reduce the searching time, we build a lookup table S_m for $\mathcal{Q}$ by uniformly quantising the K-dimensional space formed by the used K points pair measurements $\{m_k(\cdot, \cdot)\}$, i.e., the Euclidean distance, the surface normal minimal angle, the gap difference(s) in the x-, y- or z-axis for structural scenes, etc. The quantisation is achieved by uniformly dividing their corresponding value ranges into B_1, B_2, ..., B_K bins, respectively. The range of the Euclidean distance is $[d^q_{\min}, d^q_{\max}]$, where $d^q_{\min}$ and $d^q_{\max}$ denote the minimal and maximal distances of point pairs in $\mathcal{Q}$. Surface normal minimal angle falls in the range [0, π]. The gap differences fall in the ranges $[-b_x, b_x]$, $[-b_y, b_y]$ and $[-b_z, b_z]$, respectively, where b_x, b_y and b_z represent the lengths in x, y and z-axis of the minimal bounding box covering all points in $\mathcal{Q}$, respectively. Each K-dimensional bin contains all points pairs $(\mathbf{q}_i, \mathbf{q}_j) \in \mathcal{P}$ whose measurements $\{m_k(\mathbf{q}_i, \mathbf{q}_j)\}$ fall within the bin ranges.

In order to detect the matching point pairs of $(\mathbf{p}_i, \mathbf{p}_j)$, we initially evaluate the set of measurements $\{m_k(\mathbf{p}_i, \mathbf{p}_j)\}$. We then consider the thresholds $\{t_k^m\}$ associated with each measurement function in order to estimate a set of ranges $\{(m_k(\mathbf{p}_i, \mathbf{p}_j) - t_k^m, m_k(\mathbf{p}_i, \mathbf{p}_j) + t_k^m)\}$. We select all K-dimensional bins of S_m that are covered or partially covered by the estimated set of ranges and recover the associated points pairs of $\mathcal{Q}$. In particular points pairs that belong to partially covered bins are checked by verifying whether their measurements fall within the estimated set of ranges. Each extracted candidate matching pair $(\mathbf{q}_i, \mathbf{q}_j)$ is further verified by exploiting the point feature table S_f to keep only pairs whose points features correspond. In particular, we test that:

$$S_f(\mathbf{p}_i, \mathbf{q}_i) \wedge S_f(\mathbf{p}_j, \mathbf{q}_j) = 1. \tag{13}$$

Finally, using Equation 3, we evaluate the *similarity score* of each remaining candidate pair with $(\mathbf{p}_i, \mathbf{p}_j)$ and keep only the best K_p pairs. In case of very distinctive points $\mathbf{p}_i$ and $\mathbf{p}_j$, there are few correspondences in $\mathcal{Q}$

with similar local features. For such points pairs, it is more convenient to select, at first, the set of matching points using S_f and then verify each points pair using the set of measurements $\{m_k(\mathbf{p}_i, \mathbf{p}_j)\}$. This initial test is conducted by evaluating the value of $|\mathcal{M}_f(\mathbf{p}_i)| \times |\mathcal{M}_f(\mathbf{p}_j)|$, i.e., the largest number of candidate point pair matches w.r.t. $\mathbf{p}_i$ and $\mathbf{p}_j$ due to their local features. When this value is lower than a threshold, we employ this latter selection method. Our codebook-based search method allows one to efficiently range-search candidate matching point pairs using adaptive ranges for each query. If we regard the K point pair difference measurements as a K-dimensional vector, other fast search methods can be used for searching, e.g., the approximate nearest neighbor based on kd-tree [33]. However, these methods cannot handle the threshold constraints in each dimension, which may produce more candidates to test while discarding valid ones.

3.3 Iterative search of matching *N*-points bases

Finding the best corresponding point base set of a N-points base $\mathcal{B}_p \in \mathcal{P}$ requires to test $O(n^N)$ N-points bases in $\mathcal{Q}$ with a blind exhaustive search, which is often impractical due to the size of the search space. To efficiently search approximate congruent N-points bases in $\mathcal{Q}$ given a base $\mathcal{B}_p \in \mathcal{P}$, we employ an iterative approach that makes use of the codebook defined in the previous sections. We start by selecting a query set $\mathcal{B}_i$ composed by a points pair of $\mathcal{B}_p$ and search candidate congruent 2-points bases using S_f and S_m. We then iteratively add points of $\mathcal{B}_p$ to $\mathcal{B}_i$ and build the corresponding candidate congruent bases by grouping point pairs or adding single points to the previous candidate bases until $\mathcal{B}_i$ corresponds to $\mathcal{B}_p$ and all candidate bases are represented by N-points bases. Algorithm 1 describes the procedure in detail, which is also illustrated in Figure 2. In Algorithm 1, we describe two approaches to gradually expand the size of a congruent base in $\mathcal{Q}$. The first approach is to add an approximate congruent point pair having a common point with a previous base and satisfying the used congruent constraint. The second approach is to add a single point from $\mathcal{M}_f(\mathbf{p}_{i+1})$ according with the used congruent constraint. Since it is difficult to select the best approach, we use a simple strategy based on the product of set sizes $|\mathcal{M}_f(\mathbf{p}_\mathrm{i})| \times |\mathcal{M}_f(\mathbf{p}_{i+1})|$. In particular, if this size product is large, we use the former method, otherwise the latter one.

4 RANSAC pose optimization

To find the best transformation $\mathcal{T}$ that aligns the two points sets $\mathcal{P}$ and $\mathcal{Q}$, we employ a variant of the RANSAC algorithm [9], which is a widely used general technique for robust fitting of models to data corrupted with noise and outliers. The RANSAC-based alignment procedure is straightforward: randomly pick a base $\mathcal{B}_p$ of N non-collinear points from $\mathcal{P}$; detect the corresponding best congruent bases $\{\mathcal{B}_q^k\}_{k=1}^{K_g}$ and for each one compute the candidate transformation that aligns points in $\mathcal{B}_p$ with points in $\mathcal{B}_q^k$; and finally verify the recovered transformations and detect the best one using a best fit criteria. To achieve a certain probability of success, this procedure is repeated for different choices of bases from $\mathcal{P}$. Over all such trials in all iterations, we select the best transformation $\mathcal{T}$ with the best fit measure. Our

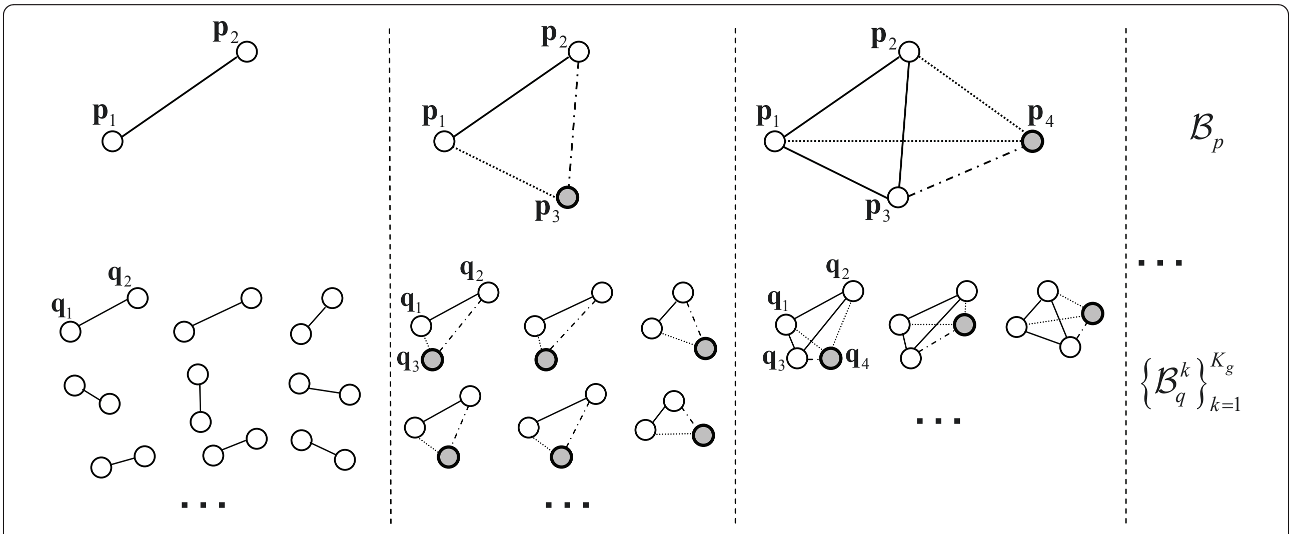


Figure 2 The procedure of *N*-points approximate congruent set searching.

adopted RANSAC algorithm terminates when one of the following two cases is reached:

1. The number of iterations reaches a predefined maximal iteration number $I_{\max}$;
2. The best transformation is not updated after I_{nou} continuous iterations.

Our method makes use of the codebook-based search scheme defined in Section 3, which is constructed before the optimization. The following sections describe in details each single step of the RANSAC iteration.

4.1 Random selection

Assumed k points in $\mathcal{P}$ having corresponding points in $\mathcal{Q}$, the probability of successfully selecting N-points from $\mathcal{P}$ having correspondences in $\mathcal{Q}$ is $p(N) \approx (k/m)^N$, where $m = |\mathcal{P}|$. To successfully recover the transformation, in general, we employ a base size $N = 3$, 4 or 5 points because the probability of success greatly decreases when the base size N increases. Moreover, to make the estimated transformation more robust, we select the N-points base $\mathcal{B}_p$ from $\mathcal{P}$ as decentralized as possible in 3-D space.

Notice that when the overlap between two scans is small only a very small subset of points in $\mathcal{P}$ have corresponding points in $\mathcal{Q}$. In this case, the probability of selecting a N-points base in $\mathcal{P}$ with a uniform distribution having a corresponding N-points base in $\mathcal{Q}$ is very low. To improve the selection probability, we propose a *probability-based RANSAC* approach described as follows. We initially build a $m \times m$ pairwise matching probability table S_p for all point pairs in $\mathcal{P}$. Given a point pair ($\mathbf{p}_i$, $\mathbf{p}_j$) in $\mathcal{P}$, its matching probability is defined by

$$S_p(\mathbf{p}_i, \mathbf{p}_j) = \frac{1}{C}\exp\left(-\varphi_1\frac{|\mathcal{M}_f(\mathbf{p}_i)|\cdot|\mathcal{M}_f(\mathbf{p}_j)|}{n^2} - \varphi_2\frac{||\mathbf{p}_i - \mathbf{p}_j||}{d^q_{\max}} - \varphi_3 s_c((\mathbf{p}_i, \mathbf{p}_j), (\tilde{\mathbf{q}}_i, \tilde{\mathbf{q}}))\right), \quad (14)$$

where C is a normalization factor, and $\{\varphi_i\}_{i=1}^3$ are three positive constants. $(\tilde{\mathbf{q}}_i, \tilde{\mathbf{q}}_j)$ is the best matched point pair of ($\mathbf{p}_i$, $\mathbf{p}_j$) with the best similarity score in Equation 3. Notice that if no approximate congruent match $(\tilde{\mathbf{q}}_i, \tilde{\mathbf{q}}_j)$ is found, we set the probability value to zero, i.e., $S_p(\mathbf{p}_i, \mathbf{p}_j) = 0$. This probability is high if:

1. Both points $\mathbf{p}_i$ and $\mathbf{p}_j$ potentially have several matches in $\mathcal{Q}$ based on their considered local descriptors (see Equation 12),
2. They are well spaced and
3. There exists a very similar 2-points base $(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{Q}$ according to the similarity measure $s_c(\cdot, \cdot)$ defined in Equation 3.

The selection of a N-points base $\mathcal{B}_p = \{\mathbf{p}_{s_k}\}_{k=1}^N$ from $\mathcal{P}$ proceeds iteratively by adding points to a selected point set $\mathcal{B}_p^c$. This is done as follows:

1. We randomly select the first two points $(\mathbf{p}_{s_1}, \mathbf{p}_{s_2})$ based on the probability values $\{S_p(\mathbf{p}_i, \mathbf{p}_j)\}_{i>j, 1 \le i, j \le m}$ of the upper triangular part of the symmetric pairwise matching probability table S_p. These points are added to the initially empty $\mathcal{B}_p^c$.
2. The next point $\mathbf{p}_{s_{k+1}}$, $k \ge 2$ is randomly selected based on the joint probability values $\left\{\prod_{\mathbf{p}_{s_k} \in \mathcal{B}_p^c} p(\mathbf{p}_i, \mathbf{p}_{s_k})\right\}_{\mathbf{p}_i \in \mathcal{P}, \mathbf{p}_i \notin \mathcal{B}_p^c}$.

In this way, there is a high probability to select a N-points base $\mathcal{B}_p$ with corresponding points in $\mathcal{Q}$.

At the end of the RANSAC iteration, if we successfully recover a candidate transformation $\mathcal{T}$ using the selected N-points base, the probability table S_p is updated. In particular, we update the corresponding elements of points $\mathbf{p}_{s_k} \in \mathcal{B}_p$ by suitably increasing the probability values: for each point $\mathbf{p}_i \in \mathcal{P}$, the new probability value $S'_p(\mathbf{p}_i, \mathbf{p}_{s_k})$ (and its symmetric value $S'_p(\mathbf{p}_{s_k}, \mathbf{p}_i)$) is evaluated as

$$S'_p(\mathbf{p}_i, \mathbf{p}_{s_k}) = S_p(\mathbf{p}_i, \mathbf{p}_{s_k}) \cdot \exp(\nu \cdot f_{\text{QLCP}}(\mathcal{T}, \delta)), \quad (15)$$

where $f_{\text{QLCP}}(\mathcal{T}, \delta)$ is the transformation fitting criterion described in Section 4.4 and v is a positive constant. The update of probabilities increases the chance to select good samples in $\mathcal{P}$, which is very useful for aligning two scans with a small overlap. To avoid unbalanced values in S_p, we decrease the probabilities of some elements if these have been updated too frequently during the RANSAC iterations. In particular, we decrease the probability value as follows:

$$S'_p(\mathbf{p}_i, \mathbf{p}_{s_k}) = S'_p(\mathbf{p}_{s_k}, \mathbf{p}_i), = \psi \cdot S_p(\mathbf{p}_i, \mathbf{p}_{s_k}), \quad (16)$$

where $\psi \in (0, 1)$ is a positive constant.

4.2 Approximate congruent group selection

After selecting an N-points base $\mathcal{B}_p$ from $\mathcal{P}$, we need to detect a set of approximate congruent N-points bases. This is done by exploiting the fast codebook structures S_f and S_m defined in Section 3. In particular following Algorithm 1, we are able to iteratively recover the set of congruent N-points bases as we select points of $\mathcal{B}_p$ from $\mathcal{P}$. We keep only the first K_g candidates according to Equation 3. These K_g N-points bases $\{\mathcal{B}_q^k\}_{k=1}^{K_g}$ are used in the following step to estimate a set of point clouds transformations.

4.3 Transformation estimation

Given two point sets $\mathcal{P}$ and $\mathcal{Q}$ with overlapping regions in arbitrary initial positions, we recover the best transformation from a prescribed family of transformations, typically rigid transformations, that best aligns the overlapping regions of $\mathcal{P}$ and $\mathcal{Q}$. In case of rigid transformation, we need a base size of at least three points to uniquely determine the aligning transformation. This means that our algorithm requires at least a pair of matching 3-points bases from $\mathcal{P}$ and $\mathcal{Q}$, respectively. In particular for any given N-points bases pairs $\mathcal{B}_p$ and $\mathcal{B}_q$, we recover the corresponding transformation $\mathcal{T}$ using the closed-form solution [34].

4.4 Transformation verification

To determine the best transformation, Aiger et al. [8] employ a *best fit* criteria called as the largest common pointset (LCP) measure $f_{\mathrm{LCP}}(\mathcal{T}, \delta)$, which refers to the transformation bringing the maximum number of points from $\mathcal{P}$ to within some δ-distance of points in $\mathcal{Q}$. Unfortunately, this criteria completely depends on the choice of the distance threshold δ. On one hand, if this threshold is too large wrong transformations may result in large LCP measure values. On the other hand, if δ is too small in some cases no transformation can be found. The main problem is that the LCP measure only considers the quantity of matched points in overlapping regions, but not their matching quality. To solve this problem to some extent, we propose to integrate a suitable matching quality measure into the LCP measure. Suppose that we have two transformations $\mathcal{T}_a$ and $\mathcal{T}_b$ computed from two different selected N-points congruent group pairs and resulting in the same LCP measure under the same distance threshold δ. Assume that the histograms of the point distances of the two transformations correspond to the ones shown in Figure 3a, b, respectively. Intuitively, $\mathcal{T}_b$ is a better solution than $\mathcal{T}_a$ because most of the corresponding point pairs have shorter distances in Figure 3b than in Figure 3a, i.e., the mean distance of the corresponding point pairs in Figure 3b is smaller than that in Figure 3a. We expect that better point matches (with shorter distances) result in a better transformation. We, thus, define a suitable matching score based on a normalized accumulated histogram $\mathcal{H}_n(\mathcal{T}, \delta)$ (see Figure 3c, d) corresponding to some given transformation $\mathcal{T}$ as follows:

$$m_s(\mathcal{T}, \delta) = \exp\left(-\lambda\left(1 - \int_0^1 \mathcal{H}_n(\mathcal{T}, \delta)\right)\right), \tag{17}$$

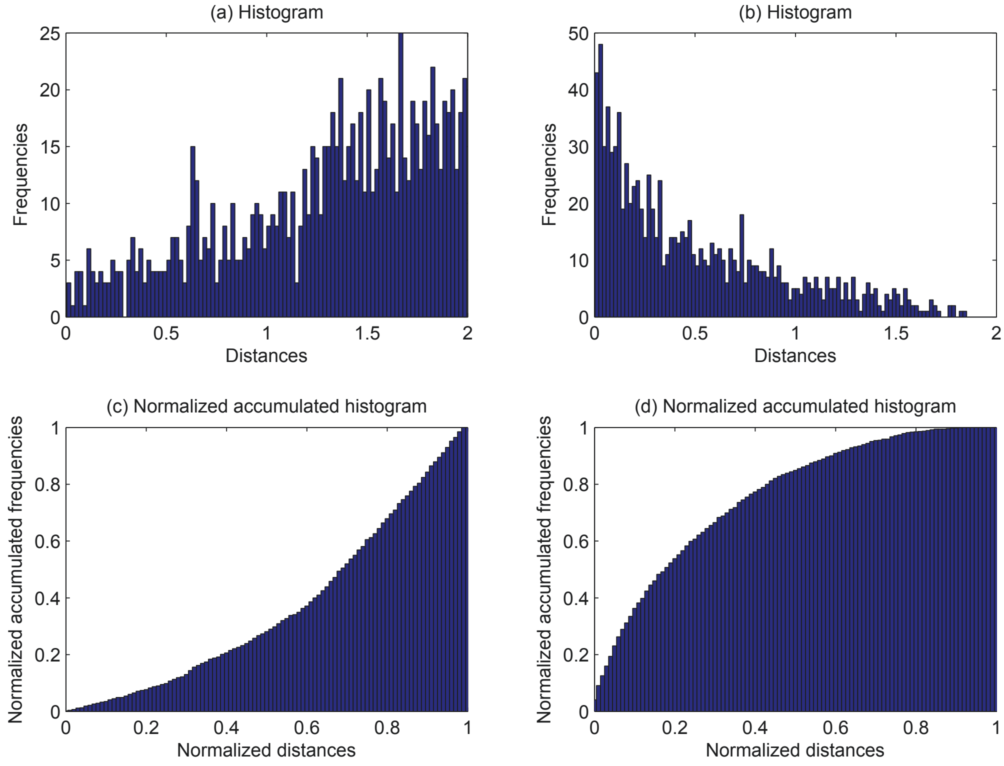


Figure 3 Histograms of points under two found transformations between two point sets resulting in the same LCP measure under the same distance threshold δ and their normalized accumulated histograms.

where λ is a positive parameter and $\int_0^1 \mathcal{H}_n(\mathcal{T},\delta)$ denotes the integral of $\mathcal{H}_n(\mathcal{T},\delta)$, i.e., the area below the cumulative curve in $\mathcal{H}_n(\mathcal{T},\delta)$. We use a quality-based LCP (QLCP) measure defined by $f_{\text{QLCP}}(\mathcal{T},\delta) = m_s(\mathcal{T},\delta) \cdot f_{\text{LCP}}(\mathcal{T},\delta)$ as our best fit criteria. By weighting $f_{\text{LCP}}(\mathcal{T},\delta)$ with the quality estimate $m_s(\mathcal{T},\delta)$ the QLCP measure is made less sensitive to the choice of δ than the LCP measure.

After the *transformation estimation* step we evaluate each recovered transformation $\mathcal{T}_k$ between $\mathcal{B}_p$ and $\mathcal{B}_q^k$ w.r.t. the mean alignment error. In particular, we test whether the error $\frac{1}{N}\sum_{\mathbf{p}_i \in \mathcal{B}_p, \mathbf{q}_i^k \in \mathcal{B}_q^k} ||\mathcal{T}_k \mathbf{p}_i - \mathbf{q}_i^k||^2$ is less than some predefined threshold, where $\mathcal{T}_k\mathbf{p}_i$ denotes the transformed point of $\mathbf{p}_i$ via $\mathcal{T}_k$. We further verify each remaining transformation by detecting how many points in $\mathcal{P}$ have correspondences in $\mathcal{Q}$ under $\mathcal{T}_k$ and then measuring the matching score as described in Equation 17. We say that one point $\mathbf{p} \in \mathcal{P}$ have a corresponding point in $\mathcal{Q}$ under $\mathcal{T}_k$, if there exists some point in $\mathcal{Q}$ closer within δ-distance to the transformed point $\mathcal{T}_k\mathbf{p}$, i.e., $\exists_{\mathbf{q} \in \mathcal{Q}} ||\mathbf{q} - \mathcal{T}_k\mathbf{p}|| \leq \delta$. For efficiency, we use the approximate nearest neighbours [33] for neighbourhood querying in $\mathbb{R}^3$. We first select a fixed number of points $\{\mathbf{p}_i\} \in \mathcal{P}$ and apply the transformation $\mathcal{T}_k$. Then, for each transformed point $\mathcal{T}_k\mathbf{p}_i$ we query the nearest neighbour in $\mathcal{Q}$. If enough points of $\{\mathbf{p}_i\}$ are matched, we perform a similar tests for the remaining points in $\mathcal{P}$ and assign to $\mathcal{T}_k$ a score based on our used QLCP measure.

Finally, we update the current best transformation found with the best QLCP measure and start the next RANSAC iteration.

5 Point sampling approaches

Given two large point sets $\mathcal{P}$ and $\mathcal{Q}$, matching approximatively congruent sets of points over the entire data set is not feasible. Thus, we need efficient point sampling strategies to quickly search corresponding sets and to effectively estimate and verify their transformation on a limited number of meaningful candidates points. The reliability of the proposed registration approach depends, to some extent, on the used sampling strategy. If we sample too many points from scarcely meaningful regions the registration might converge slowly, find the wrong transformation (such as solutions showing sliding effects produced by samples poorly constraining the transformation), or even diverge, especially in the presence of noise and outliers. Several point sampling techniques for point cloud alignment were recently proposed [6,35-37]. In [6] the random, uniform (over the surface area of a model) and normal-space sampling are considered to evaluate the convergence of the ICP algorithm. The normal-space sampling algorithm tries to uniformly spread the normals of the selected points on the sphere of directions. The aim is to consider points that sufficiently constrain the estimated rigid transformation and improve the alignment quality by reducing surface sliding effects. Gelfand et al. [35] proposed a variant of this algorithm to make the transformation estimation geometrically stable by selecting points that reduce both translational and rotational uncertainties in the ICP algorithm. This technique samples points in order to equally constrain all eigenvectors of the covariance matrix estimated from the points and the normals of the overlapping region of two point clouds. A similar approach was used in [36] to conceive a probability function to guide the selection of stable sample points, which are also constrained by specific features. Torsello et al. [37] proposed a sampling technique to select feature points with high-local distinctiveness, which is inversely proportional to the average local radius of curvature and related to the area formed by similar points in the neighbourhood of each point. Nehab and Shilane [38] discussed the limitation of the area-based uniform sampling, where the probability of a surface point being sampled is equal for all surface points. This type of sampling might produce points very closed to each other and miss important surface features, which could successfully constrain the transformation. To overcome these drawbacks they proposed a stratified point sampling strategy ensuring an even distribution of the sample points on all surface, which implies a higher probability to catch important surface regions. This algorithm uses the voxelization of the model to generate random samples with controlled intra-distances. Other sampling strategies providing with uniform distribution of sample points on a surface are based on the farthest point [39] and Poisson disk sampling [40], which require the computation of geodesic distances.

In this section, we investigate four sampling approaches: random, uniform, probabilistic and combined sampling, which are described in details in the following.

5.1 Random sampling

Random sampling is the simplest and widely used sampling technique. In random sampling, each item or element of the population has an equal chance of being selected at each draw. A sample is random if the method for obtaining the sample meets the criterion of randomness (each element having an equal chance at each draw). The actual composition of the sample itself does not determine whether or not it was a random sample.

5.2 Uniform sampling

To achieve a high probability of success for registering two overlapping point sets, we expect that the sampled point sets have similar point densities. If the acquired surfaces present similar point densities, the above mentioned random sampling deserves to be an acceptable choice, which will also result in similar point densities in the sampled overlapping parts. However, this assumption does not hold in general. Point density depends on the distances and on the incident angles of the scanned surfaces with respect to the scanner sensor position and orientation, respectively. Normally, short distances and small incident angles lead to surfaces with high-point densities and accuracies, which we consider as high-resolution regions. Random sampling does not guarantee an equal spread of the generated points neither on the surface nor in the volume of the scanned model, and can sample points very close to each other. Thus, it is more likely that it misses important surface features than an evenly distributed sampling [38]. This effect is particularly evident in case of scanned data with non-uniform point densities.

Many approaches to perform a sampling of uniformly distributed points on the model surface can be employed [38-40]. We propose a simple and efficient variant of the method presented in [38], which is based on cubic voxelization of point clouds and that provides with samples evenly displaced on typical scanned surfaces. Given a point set $\mathcal{P}$, we can assign them into a set of 3-D cubic voxels of equal sizes, which partition the 3-D space. For each such voxel, we select the closest point to the voxel center as the sampled point. To obtain a sampled point set of a given size N_s, we start by splitting the minimal bounding box of the point cloud into a small set of 3-D cubic voxels. We then iteratively split each voxel into eight small voxels until there are enough sampled points found. With this strategy, however, we cannot obtain an accurately fixed-size set of sampled points since most of the voxels do not contain points. Let N_l be the number of sampled points at the lth level. The final level L is such that its number of points N_L is not less than N_s and the number of points at the previous level is $N_{L\text{-}1} < N_s$. To obtain a sampling of the expected size N_s, the simple way is to randomly select N_s points from the N_L points of the L level. To achieve a more uniform distribution, we propose, instead, to re-split all points in $\mathcal{P}$ into a set of cubic voxels of size $\left(\frac{8N_s^2}{N_{L-1}N_L}\right)^{\frac{1}{3}} S_{L-1}$, where $S_{L\text{-}1}$ denotes the voxel size at the (L - 1)th level. In this way, the number of uniformly sampled points is very close to N_s, but still not exact. If the number is larger than N_s, we randomly select N_s points from them. Otherwise, we add some new points from sampled point set at the Lth level.

5.3 Probabilistic sampling

If the acquired surfaces are very similar in structure, e.g., the surfaces of an indoor environment composed of a main flat wall and several small objects in the front of it, the above two sampling approaches may not be efficient for our proposed registration. One reason is that we select the best transformation based on the degree of overlap in the point sets, but not based on the whole scene structure. In the above example, points from the main single wall weakly constrain translations and generate sliding effects in the final alignment, as already discussed in [6,35-37]. Another reason is that a selected N-points base from the wall would have a large number of approximate congruent bases. This would require expensive searches, tests, estimates and verifications of candidate transformations over a large set of approximate congruent bases. To avoid these problems, we expect to consider more points from objects in front of the wall, which would reduce the computation and better constrain the rigid transformation. Similarly to [37], this is achieved by utilizing a probabilistic sampling technique, which selects points based on their likelihoods computed from a specific weight function. The weight function determines how much each point is relevant for registration and is basically associated with the local geometrical properties of the surface at each point. We experimented with two different weight functions ω_{SV} and ω_{APD} based on surface variation and adjacent point distance, respectively.

The *surface variation* of a point $\mathbf{p}$ is defined as:

$$\omega_{\text{SV}}(\mathbf{p}) = \frac{3\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3}, \tag{18}$$

where $\lambda_1 \leq \lambda_2 \leq \lambda_3$ are the eigenvalues corresponding to the principal components of a set of k points in the neighbourhood of $\mathbf{p}$. $\omega_{\text{SV}}(\mathbf{p}) \in [0, 1]$ indicates how much the surface at $\mathbf{p}$ locally deviates from the tangent plane [41]. In practice, $\omega_{\text{SV}}(\mathbf{p})$ roughly approximates the mean curvature at $\mathbf{p}$: when its value is close to zero it indicates that the surface is locally planar, while, when $\omega_{\text{SV}}(\mathbf{p})$ is large, $\mathbf{p}$ identifies an interesting feature like corners, bumps, etc.

The *adjacent point distance* [42] is defined by exploiting the grid structure of range image $\mathcal{I}$ related to the acquired point cloud. Let $\mathbf{p}$ be a valid point associated with a pixel of $\mathcal{I}$, its adjacent point distance $\mathcal{A}_d(\mathbf{p})$ is defined as the median of the distances between $\mathbf{p}$ and its adjacent valid points $\mathbf{p}_k$ in a 3 × 3 neighbourhood of $\mathbf{p}$, i.e., $\mathcal{A}_d(\mathbf{p}) = \text{median}_k\|\mathbf{p} - \mathbf{p}_k\|$. Then, to reduce the effect of measurement noise, a median filter of size 5 × 5 is applied on the resulting adjacent point distance map $\mathcal{A}_d$ to get a filtered map $\tilde{\mathcal{A}}_d$. The weight function ω_{APD} of a point $\mathbf{p}$ is then defined as:

$$\omega_{\text{APD}}(\mathbf{p}) = \begin{cases} \hat{\mathcal{A}}_d^2 & \text{if } \hat{\mathcal{A}}_d(\mathbf{p}) \geq \hat{\mathcal{A}}_d, \\ \hat{\mathcal{A}}_d(\text{x})^2 & \text{otherwise.} \end{cases} \tag{19}$$

where $\hat{\mathcal{A}}_d$ denotes the 95th percentile of the adjacent point distances in an ascending order of all points in $\tilde{\mathcal{A}}_d$. The use of $\hat{\mathcal{A}}_d$ effectively suppresses estimation errors from outliers. $\omega_{\mathrm{APD}}(\mathbf{p})$ estimates the local sampling sparsity of the scanned surface at $\mathbf{p}$. High values of ω_{APD} characterize those points having a neighbourhood sparsely sampled, typical of corner and edge points and regions scanned with low incident angle, which are likely located in the overlapping area of the models.

5.4 Coupled sampling

Besides the aforementioned three sampling approaches, we also consider to couple different sampling approaches in some order. For example, a point set $\mathcal{P}_1$ is selected from the initial point set $\mathcal{P}$ based on probabilistic sampling, after that, another point set $\mathcal{P}_2$ is selected from $\mathcal{P}_1$ based on uniform sampling. The finally sampled point set can be selected from the initial point set $\mathcal{P}$ via two or more sampling in some order with given sampling ratios. The sampling ratios of the finally sampled point set $\mathcal{P}_S$ from $\mathcal{P}$ via S sampling processes are denoted as $|\mathcal{P}_1| : |\mathcal{P}_2| : \cdots : |\mathcal{P}_S|$ where $|\cdot|$ denotes the set size. For different scans to be aligned, we can select a suitable sampling approach.

6 Integrating texture features

If the acquired models lack geometric details to be used as good anchor points for correct registration, we can still employ features extracted from other visual sources provided by the laser range scanners, e.g., the reflectance image. Laser range scanners are non-contact 3-D scanners that measure the distance from the sensor to points in the scene, typically in a regular grid pattern. A natural byproduct of this acquisition process is the *reflectance image*. A reflectance image (shown in Figure 4) stores in each pixel the portion of laser light reflected from the corresponding surface point, providing with important information about its texture. Both 3-D space distribution and texture characteristics of the texture features extracted from the reflectance images can well constrain a rigid transformation in 3-D space. As shown in [11,12] texture features can be effectively used to identify anchor points leading to a well-constrained rigid transformation.

A feature is accompanied by a descriptor, which locally and compactly describes the texture around the feature pixel. In our application, we are interested in good local feature descriptors, which should have a high-local distinctiveness, invariant w.r.t. affine

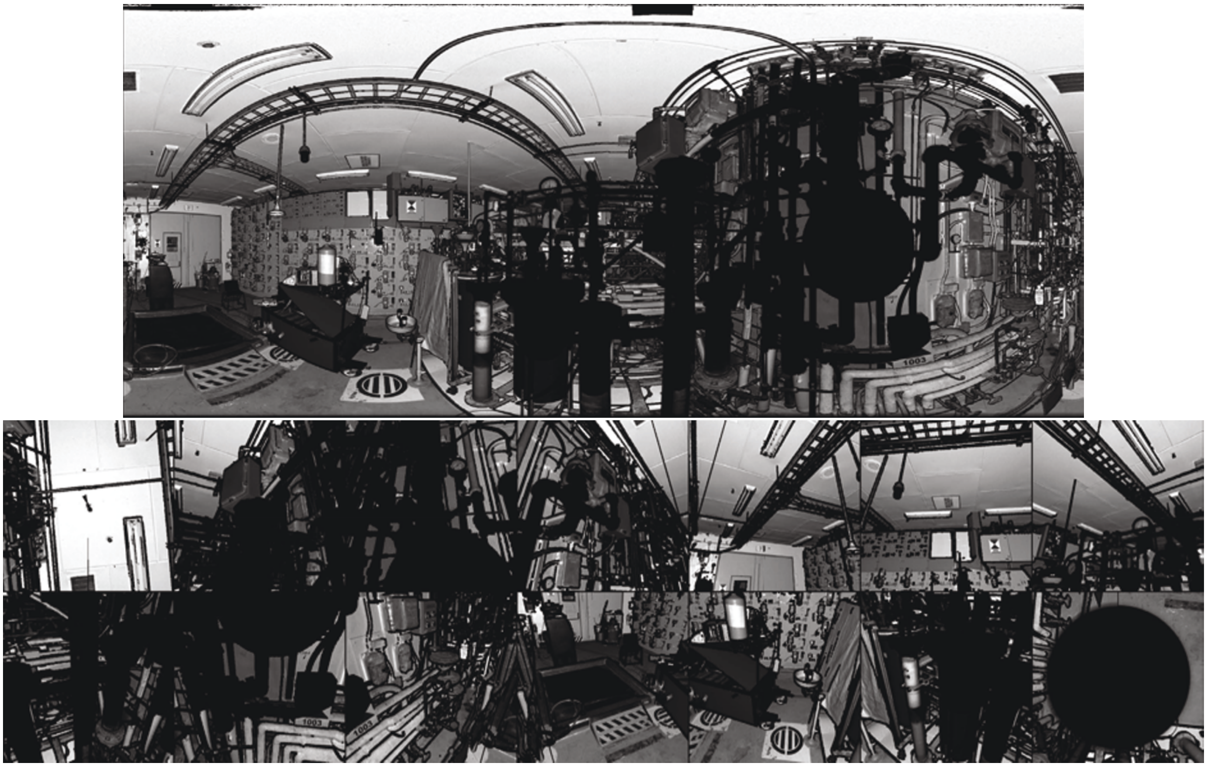


Figure 4 Atlas of rectified images generated from a reflectance image: (*top*) reflectance image; (*bottom*) atlas of 14 rectified images generated by sampling a sphere every 60°.

transformations, and possibly robust w.r.t. illumination changes and local deformation. Several local feature descriptors were presented in the literature [32,43]. Among them, the most suitable for our application are the SIFT [44], SURF [45] and FAST [46] descriptors, which we extract from the reflectance images of our scanned models and consider as relevant sample points to be matched by our registration algorithm.

6.1 Reflectance image rectification

The aforementioned features cannot be efficiently extracted directly from the reflectance image associated with a scanned model, since its intrinsic spherical format strongly affects the quality of their descriptors, which are not designed to be robust w.r.t. spherical distortions. Indeed, typical laser scanner acquisition systems are usually composed by a fixed platform and a rotating head, which are naturally modelled by simple spherical projections. The resulting acquired images are then obtained by mapping spherical images onto single image planes. The meridians of a spherical image are mapped to vertical viewing planes, and the parallels are mapped to viewing cones with the vertex in the sensor position.

In order to reduce the distortion induced by the spherical projection, we compute the above mentioned feature descriptors on an atlas of rectified images (shown in Figure 4). This is possible since both the spherical projection and the atlas perspective projections will share the same point of view.

To recover the set of rectified images, we initially select a field of view value α_{fov} (in our experiments $\alpha_{\text{fov}} = 60°$). We then calculate the width w_r and height h_r of each rectified image by constraining the pixel resolution to the resolution of the spherical image equator, i.e., the width w_s of the spherical image $: w_r = h_r = w_s \cdot \frac{\alpha_{\text{fov}}}{360^\circ}$. Given the pixel dimensions of the image plane, we define a standard perspective projection whose principal point is represented by the central pixel of the image plane. The only missing intrinsic parameter, the projection focus, can be easily recovered given α_{fov}, w_r and h_r. To determine the extrinsic parameters of each projection, we fix the camera point of view to the spherical projection point of view, i.e., the local origin of the point cloud. We then sample the sphere with v equally distributed directions. The value of v depends on the required field of view and is estimated such that the images contained in the atlas completely cover the initial spherical image. Given the camera direction d_i we recover the remaining extrinsic parameters of the ith image aligning the camera principal direction to d_i and the vertical direction with the vertical direction of the spherical image. Exploiting each projection matrix, we can associate to each final image pixel its corresponding viewing ray, and from this the pixel's corresponding coordinates in the original spherical image. These correspondences are used to perform a bilinear interpolation of the spherical image to recover each single finally rectified image (see Figure 4).

The atlas generation only depends on the field of view used. Small values generate multiple small images, whereas large values generate fewer images but with higher-perspective distortions.

6.2 Texture features extraction and integration

From the original reflectance image or the atlas of rectified images generated from a reflectance image, we extract a set of texture features for registration by using the following methods:

1. SIFT [44]: this feature descriptor encodes the trend of the image local gradient around a pixel as a histogram of typically 128 bins. This descriptor is invariant w.r.t. scaling and rigid 2-D transformation and robust w.r.t. affine distortion, addition of noise, and change in illumination. This descriptor is very accurate in identifying relevant interest point, but its computation is usually slow without exploiting the GPU [47].
2. SURF [45]: this method efficiently detects features by computing a rough approximation of the Hessian matrix using integral images. The resulting descriptor is based on sums of approximated 2-D Haar wavelet responses, which is more compact and much faster to compute than the SIFT descriptor.
3. FAST [48]: this techniques classifies a pixel as corner if there is a sufficiently large set of relevantly brighter (darker) pixels in a circular pixel neighbourhood of fixed radius. This feature detection algorithm is very fast, up to 30 times faster than the SIFT one, but is not invariant w.r.t. scaling and does not provide with effective descriptors [43], which are usually computed by using other techniques (e.g., with the SURF method in this paper).
4. Harris corner detector [49]: it has been widely used in image processing and computer vision, and computes corner features by analysing the local changes of the image intensity with patches shifted by a small amount in different directions. It is not scale and affine invariant, and usually generates a high number of features.

The 3-D points corresponding to the extracted texture features are then considered as sampled points and used with their descriptors by our registration algorithm to match congruent sets of points, as described in Sections 2 and 3.

7 Experimental results

7.1 Test data and evaluation criteria

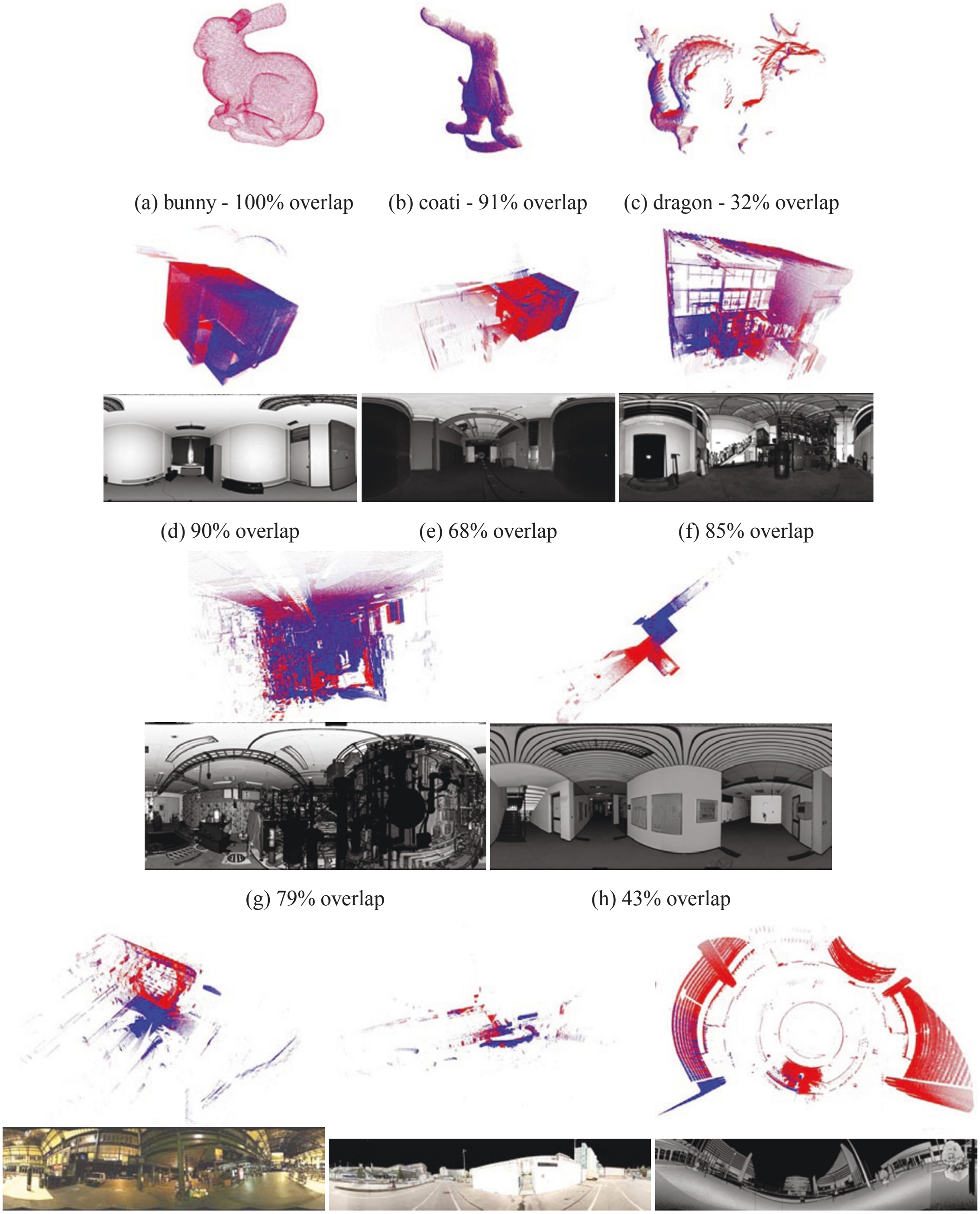
We tested our point-based registration algorithm on a variety of input data with varying amount of noise, outliers, and extent of overlap. Our test dataset includes some small object models as shown in Figure 5a, b, c, which are selected from the data provided in the demo application of the 4PCS algorithm [8]^{b}. Other test data are models of large indoor/outdoor scenes acquired by different types of scanners, which are mostly selected from [20], as shown in Figure 5d, e, f, g, h, i, j, k. In total, we tested 11 pairs of scans with different extents of overlap as shown in Figure 5. These 11 pairs of scans are noted by $\{\mathcal{G}_n\}_{n=\mathrm{a},\ldots,\mathrm{k}}$ corresponding to the models shown in Figure 5a, b, c, d, e, f, g, h, i, j, k, respectively. The small objects models in $\mathcal{G}_\mathrm{a} - \mathcal{G}_\mathrm{c}$ consist of around 20,000-30,000 points. Indoor scan data $(\mathcal{G}_\mathrm{d} - \mathcal{G}_\mathrm{i})$ were captured by Z+F IMAGER 5003/5006/5006i laser rangescanners. Outdoor scan data ($\mathcal{G}_\mathrm{j}$and $\mathcal{G}_\mathrm{k}$) were captured by the RIEGL LMS-Z420i laser range-scanner. The accuracy of a point acquired by the Z+F IMAGER 5003 (5006 and 5006i) laser scanner is 3 mm (1 mm) along the laser-beam direction at a maximal distance of 50 m from the scanner. The accuracy of the RIEGL LMS-Z420i laser scanner is 10 mm at 50 m. The resolution of all indoor scans is about 2, 530 × 1, 080 and the resolution of all outdoor scans is 3, 000 × 666. No surface normals were provided for points in the scan data $\mathcal{G}_\mathrm{a}$, $\mathcal{G}_\mathrm{b}$ and $\mathcal{G}_\mathrm{c}$. For the other scans, we always employed the surface normals into our registration algorithm. The scan data in $\mathcal{G}_\mathrm{a} - \mathcal{G}_\mathrm{c}$ were scaled so that the bounding box diagonal lengths of the first scans in these scan pairs are taken as 100 units. The bounding box diagonal lengths of the first scans of other eight scan pairs $\mathcal{G}_\mathrm{d} - \mathcal{G}_\mathrm{k}$ are around 10, 28, 26, 37, 77, 142, 1, 860, and 398 *m*, respectively. The overlap rates shown in Figure 5 were computed as follows. The overlap rate of two point sets $\mathcal{P}$ and $\mathcal{Q}$ is defined as $o(\mathcal{P}, \mathcal{Q}) = \min\left(\frac{|\mathcal{P}\in\mathcal{Q}|}{|\mathcal{P}|}, \frac{|\mathcal{Q}\in\mathcal{P}|}{|\mathcal{Q}|}\right)$, where $|\cdot|$ denotes the size of a point set and $\mathcal{P} \in \mathcal{Q}$ denotes a subset of $\mathcal{P}$ in which for each point there exist at least one point in $\mathcal{Q}$ whose distance to it is below a given threshold. To produce a reasonable overlap rate in 3-D space, we compute the overlap rate by using large point subsets selected from $\mathcal{P}$ and $\mathcal{Q}$ using our voxel-based uniform sampler instead of using original point sets. Our registration algorithm was implemented in C++ on a Windows XP system and integrated into our commercial software *JRC 3-D Reconstructor*. All experiments were executed on a 2.67 GHz Intel machine.

We fixed the poses of all reference scans (i.e., $\{\mathcal{Q}\}$) to an identity rotation matrix and no translation. To evaluate the transformation estimation accuracy of our registration algorithm, we first employed our algorithm on all tested pairs of scans to recover a good initial transformation for each scan pair and then applied the ICP optimization algorithm [5] to get a well-aligned transformation. We observed that the mean residual error after the ICP registration optimization was always comparable with the laser scanner measurement error, which is much lower than the estimation errors of our proposed *N*-points congruent sets (NPCS) registration algorithm. For this reason, we regarded this ICP-optimized transformation as the ground truth transformation $\mathcal{T}_g$ for evaluation. Thus, given an estimated transformation $\mathcal{T}$ from $\mathcal{P}$ to $\mathcal{Q}$, the estimation error is defined as the median of the point distances after applying $\mathcal{T}$ and $\mathcal{T}_g$ onto $\mathcal{P}$, i.e., $\mathrm{median}_{\mathbf{p}\in\mathcal{P}}||\mathcal{T}\mathbf{p} - \mathcal{T}_g\mathbf{p}||$.

The transformation estimation accuracy was statistically evaluated by running our registration algorithm N_run = 20 times on each tested pair of scans. In each run, we refresh the input data by setting a random pose for the moving scan followed by re-sampling. For each scan pair, we set a suitable maximal estimation error Δ_max in advance. If the estimation error was above Δ_max, we considered it as failed. The maximal estimation errors were set as Δ_max = 5 units, Δ_max = 1 m, Δ_max = 2 m and Δ_max = 5 m for small object models $\mathcal{G}_\mathrm{a} - \mathcal{G}_\mathrm{c}$, the indoor models $\mathcal{G}_\mathrm{d} - \mathcal{G}_\mathrm{h}$, the large indoor models $\mathcal{G}_\mathrm{i}$ and the outdoor models $\mathcal{G}_\mathrm{j} - \mathcal{G}_\mathrm{k}$, respectively. Based on the number N_suc of successful estimations and the number N_run of runs the following three indicators were then used to evaluate our method: (1) the *successful estimation rate* $S_r = \frac{N_\mathrm{suc}}{N_\mathrm{run}}$ to evaluate its robustness; (2) the *median estimation error* Δ among all N_suc successful estimations to evaluate its accuracy; (3) the *median estimation time t* over all N_run runs to evaluate its efficiency. Note that the estimation time does not include pre-processing computation time (i.e., texture feature detection/matching, point sampling, etc), but it incorporates the codebook building time, which took <1 s with the basic RANSAC and around two seconds with the probabilistic RANSAC using our parameter setting. The feature detectors listed by increasing processing time are Harris, FAST, SURF and SIFT. The point sampling approaches listed by increasing processing time are random, probabilistic and uniform samplings.

7.2 Performance evaluation

The performance of our proposed NPCS registration algorithm was evaluated on the aforementioned test data. The main parameters were set as follows. We employed congruent points sets of size N = 4. The two main parameters for searching best *N*-points congruent sets in Algorithm 1 were set as K_p = 2, 000 and K_g = 50. The exponential parameter λ = 1 for computing our



(a) bunny - 100% overlap (b) coati - 91% overlap (c) dragon - 32% overlap

(d) 90% overlap (e) 68% overlap (f) 85% overlap

(g) 79% overlap (h) 43% overlap

(i) 45% overlap (j) 39% overlap (k) 12% overlap

Figure 5 All test pairs of scan data described with aligned point clouds (*different colours* represent different scans), approximated overlapping percentage, and one reflectance image for each pair of scans representing: a, b, c small objects; d, e, f, g, h indoor scenes; i large indoor scenes; j, k outdoor scenes.

proposed QLCP measure in Equation 17. Given two point sets $\mathcal{P}$ and Q, we tried to recover the transformation from $\mathcal{P}$ (moving point set) to $\mathcal{Q}$ (reference point set). We selected 500 points from $\mathcal{P}$ and 1,000 points from $\mathcal{Q}$ for searching N-points congruent sets between them. However, for transformation verification, we selected larger subsets, i.e., 1,000 points from $\mathcal{P}$ and 2,000 points from $\mathcal{Q}$. This allows the estimation of a more accurate and robust transformation. The voxel-based uniform sampling approach was used to select these points. Our registration algorithm always used the surface normals of points when provided. The maximal normal deviation for the corresponding point pairs was set to 30°. In our experiments, we used the RANSAC-based approach by setting $I_{\max} = 1,000$ and $I_{\text{nou}} = 200$, which are the allowed maximal iteration number and the continuous iteration number in which no better transformation was found, respectively. The estimation errors of the first three scan pairs $\mathcal{G}_{\text{a}} - \mathcal{G}_{\text{c}}$ were reported in canonical units (i.e., 100 units are equal to the bounding box diagonal lengths), while those of the other eight scan pairs $\mathcal{G}_{\text{d}} - \mathcal{G}_{\text{k}}$ in meters. Notice that the same parameter values were used in all experiments described below, unless clearly stated otherwise for particular experiments.

Figure 6 shows the performance comparison of our NPCS algorithm with different K_p and K_g on three scan pairs $\mathcal{G}_{\text{a}}$, $\mathcal{G}_{\text{e}}$ and $\mathcal{G}_{\text{i}}$. First, we fixed $K_g = 50$ and tested the effects of different K_p on the registration accuracy (median estimation errors), efficiency (median estimation times) and robustness (successful estimation rates), as shown in Figure 6a, b, c, respectively. We can observe that larger values of K_p led to an improvement of the registration accuracy and robustness, but also required longer execution times. In Figure 6c, we can notice that 100% successful estimation rates were achieved for all tested scan pairs when $K_p \geq 1,000$. Second, we fixed $K_p = 2,000$ and tested the effects of different K_g on registration performance shown in Figure 6d, e, f. The variation of K_g results in similar effects as for K_p. When $K_g \geq 50$, the successful estimation rates were always $S_r = 100\%$ for all tested scan pairs.

Table 1 illustrates the performance evaluation of our NPCS algorithm on three scan pairs $\mathcal{G}_{\text{c}}$, $\mathcal{G}_{\text{d}}$ and $\mathcal{G}_{\text{j}}$ when the size N of congruent point sets was set as $N = 3, ..., 6$, respectively. As explained before, large values of N (≥ 6) result in a low probability of successfully selecting N-points congruent sets between two point sets and in longer estimation times. Small values of N ($= 3$) led to an increase of the candidate N-points congruent sets in $\mathcal{Q}$ given N points in $\mathcal{P}$, but sometimes the matched sets found are not enough to well constraint the transformation and the algorithm falls into local minima.

The performance comparison of different sampling approaches on five scan pairs is shown in Table 2. In this experiment, we tested the following sampling strategies: the *random* sampling; the voxel-based *uniform* sampling; two probabilistic sampling approaches (*probAPD*-based on adjacent point distances, *probSurfVar*-based on surface variations); four texture feature-based sampling using the SIFT, SURF, FAST and HARRIS feature detectors as samplers. Some sampling strategies

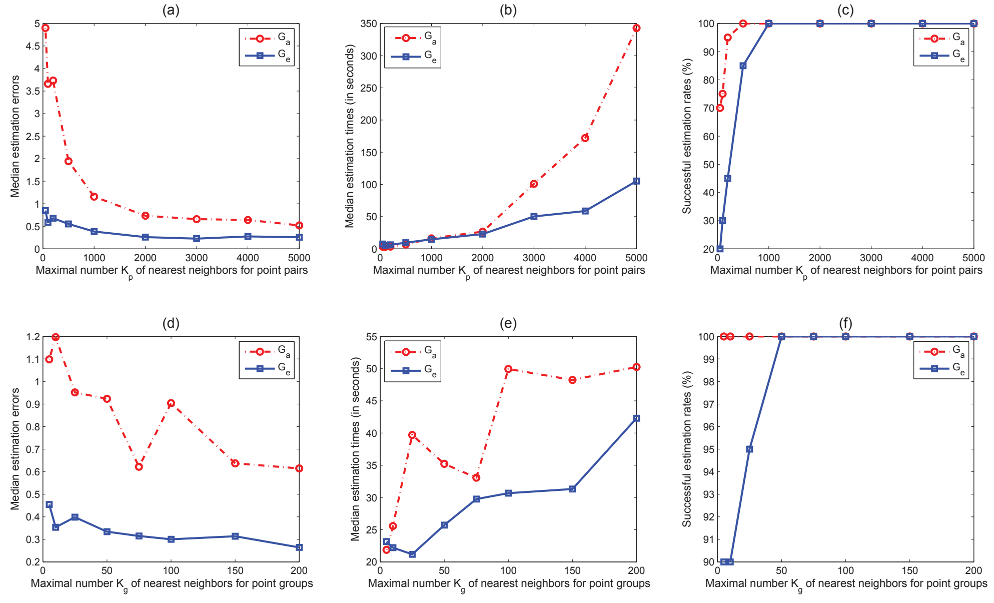


Figure 6 Performance comparison with different K_p and K_g on three scan pairs $\mathcal{G}_{\text{a}}$, $\mathcal{G}_{\text{e}}$ and $\mathcal{G}_{\text{i}}$.

Table 1 Performance evaluation of our NPCS algorithm with different sizes *N* of congruent point sets

N	$\mathcal{G}_{\mathrm{c}}$			$\mathcal{G}_{\mathrm{d}}$			$\mathcal{G}_{\mathrm{j}}$		
	Δ	S_r (%)	*t* (s)	Δ	S_r (%)	*t* (s)	Δ	*Sr*(%)	*t* (s)
3	0.7060	100	61.5	0.0770	100	29.4	0.5903	100	18.5
4	0.3906	100	35.1	0.0508	100	45.4	0.7819	100	16.6
5	0.4532	100	39.2	0.0484	100	53.9	0.6114	100	20.6
6	0.4277	100	40.8	0.0462	100	90.6	0.9586	95	77.1

were coupled and reported in Table 2 combined with the character '+'. For instance, the coupled sampling "random+uniform" denotes that we first applied the random sampling to get a point subset $\mathcal{P}_1$ from the initial point set $\mathcal{P}$ and then applied the uniform sampling to get the final point subset $\mathcal{P}_2$ from $\mathcal{P}_1$. Here, the sampling ratios of all coupled sampling approaches were set to $\mathcal{P}_2 : \mathcal{P}_1 = 0.5$. All texture feature-based sampling methods were coupled with random or uniform sampling. From Table 2, we observe that the uniform sampling performs better than the random sampling both when is considered alone and when is coupled with other sampling strategies, i.e., with the probAPD-based sampling and feature detectors. Among the probabilistic sampling methods the probAPD-based sampling resulted to be a very good choice if available, since it relevantly improved the registration robustness and also increased its accuracy. On the contrary, the probSurfVar-based sampling showed lack of robustness, but worked very well for models rich of geometric features, which were captured from geometrically complex scenes, as those in $\mathcal{G}_{\mathrm{g}}$. In some scenes, texture feature detectors turned out to be a good first sampler. In those cases, they improved the transformation accuracy as shown by 'FAST+uniform" in $\mathcal{G}_{\mathrm{e}}(\Delta = 0.0672\ \mathrm{m})$ and by "HARRIS+random" in $\mathcal{G}_{\mathrm{g}}(\Delta = 0.0121\ \mathrm{m})$ and $\mathcal{G}_{\mathrm{i}}(\Delta = 0.0820\ \mathrm{m})$. Among the texture feature-based sampling "SIFT+uniform" and "FAST+uniform" also demonstrated a very good robustness by always scoring $S_r = 100\%$ as the probAPD-based sampling, but with improved accuracies for all models apart from those in $\mathcal{G}_{\mathrm{j}}$. Notice that if it is not possible to extract enough texture features the remaining points were selected by the following coupled sampling approach.

Our employed QLCP measure for verifying the estimated transformation depends on two main parameters, i.e., λ in Equation 17 and δ-distance. In this paper, we computed $\delta = \alpha\dot{d}_{v\mathrm{ref}}$, where α is a positive constant and $\dot{d}_{v\mathrm{ref}} = \mathrm{median}_{\mathbf{q}\in\mathcal{Q}_v}||\mathbf{q} - n_c(\mathbf{q})||$, where $\mathcal{Q}_v \subset \mathcal{Q}$ represents the set of points used for transformation verification and $n_c(\mathbf{q})$ denotes the closest point of $\mathbf{q}$ in $\mathcal{Q}_v$. Table 3 shows the comparison of our NPCS algorithm when using the LCP or the QLCP measure with different parameters on three scan pairs: $\mathcal{G}_{\mathrm{b}}$, $\mathcal{G}_{\mathrm{i}}$, and $\mathcal{G}_{\mathrm{j}}$. In practice, the LCP measure corresponds to the QLCP measure with $\lambda = 0$. From Table 3, we observe that our proposed QLCP measure led to higher successful estimation rates and more accurate transformations than the LCP one. In addition, notice that a large λ slightly increased the estimation robustness when a large α was used. In the rest of the experiments reported in the paper, we used $\lambda = 1$ and $\alpha = 2$.

Table 2 Performance evaluation of different sampling approaches on five pairs of scans

Sampling approaches	$\mathcal{G}_{\mathrm{c}}$		$\mathcal{G}_{\mathrm{e}}$		$\mathcal{G}_{\mathrm{g}}$		$\mathcal{G}_{\mathrm{i}}$		$\mathcal{G}_{\mathrm{j}}$	
	Δ	S_r (%)	Δ	S_r (%)	Δ	S_r (%)	Δ	S_r (%)	Δ	S_r (%)
Random	0.4561	100	★	0	0.0606	95	★	0	★	0
Uniform	0.4465	100	0.2160	95	0.0854	100	0.5146	95	0.4231	100
probAPD			0.2045	100	0.0501	100	0.3260	100	0.3548	100
probSurfVar			0.0774	10	0.0329	100	★	0	★	0
Random+uniform	0.4719	100	0.1547	100	0.0410	100	0.1961	5	★	0
Random+probAPD			0.1670	100	0.0679	100	0.4719	65	0.8053	45
Uniform+probAPD			0.2001	100	0.0629	100	0.4211	100	0.4343	100
SIFT+random			0.1020	100	0.0276	100	0.6754	5	★	0
SURF+random			0.0840	85	0.0264	100	0.3329	10	0.2675	30
FAST+random			0.1331	75	0.0252	100	0.4970	55	★	0
HARRIS+random			0.0679	100	0.0121	100	0.0820	100	0.3756	100
SIFT+uniform			0.0647	100	0.0349	100	0.1491	100	0.1959	100
SURF+uniform			0.0848	100	0.0542	100	0.3064	100	0.3426	100
FAST+uniform			0.0672	100	0.0446	100	0.1118	100	0.2719	100
HARRIS+uniform			0.1131	100	0.0194	100	0.1077	100	0.5959	100

"★" denotes that no correct transformations were found in 20 runs

Table 3 Performance comparison between the LCP and QLCP measures on different models

	$\mathcal{G}_\mathrm{b}$			$\mathcal{G}_\mathrm{i}$			$\mathcal{G}_\mathrm{j}$		
	Δ	***S**_r* **(%)**	***t* (s)**	**Δ**	***S**_r* **(%)**	***t* (s)**	**Δ**	***Sr*(%)**	***t* (s)**
LCP (α = 1)	0.3738	100	21.4	0.7377	95	25.6	0.7141	100	6.6
LCP (α = 2)	0.7252	100	8.3	0.7264	90	26.5	2.4107	100	15.2
LCP (α = 4)	1.2058	100	16.8	0.9550	30	24.8	3.7339	10	27.1
LCP (α = 8)	0.5809	100	47.9	1.3048	20	22.6	4.5258	10	37.8
QLCP (λ = 1, α = 1)	0.3524	100	31.3	0.5304	95	33.2	0.4041	100	11.7
QLCP (λ = 1, α = 2)	0.2622	100	30.8	0.5225	100	43.0	0.5107	100	14.6
QLCP (λ = 1, α = 4)	0.2895	100	62.0	0.7329	95	43.9	0.7857	100	28.6
QLCP (λ = 1, α = 8)	0.3934	100	81.5	0.8767	80	85.1	0.9193	100	97.1
QLCP (λ = 2, α = 1)	0.2127	100	49.3	0.3264	95	29.1	0.3782	100	10.9
QLCP (λ = 2, α = 2)	0.3055	100	35.9	0.4783	100	61.2	0.5394	100	23.5
QLCP (λ = 2, α = 4)	0.2904	100	59.3	0.4910	100	97.8	0.6343	100	33.6
QLCP (λ = 2, α = 8)	0.3537	100	142.3	0.8897	85	79.5	0.7970	100	64.0

The integration of scene structure information to our NPCS algorithm can greatly reduce the estimation time, as described in Section 2 and illustrated in Table 4. This can also slightly improve the accuracy of the recovered transformation. Here, the scans in $\mathcal{G}_\mathrm{d}$ and $\mathcal{G}_\mathrm{f}$ were automatically classified as orthogonal scenes.

Table 5 shows the performance evaluation of our NPCS algorithm on $\mathcal{G}_\mathrm{g}$ and $\mathcal{G}_\mathrm{h}$ using texture features detected from original reflectance images and from atlases of rectified reflectance images. In this experiment the texture features were used to first sample the models and their descriptors were used to improve the matching of congruent sets, as described in Section 2. These techniques were compared with the pure uniform sampling method, which worked well in $\mathcal{G}_\mathrm{g}(S_r = 100\%)$, but failed completely in $\mathcal{G}_\mathrm{h}(S_r = 0\%)$.

The reason of complete failure in $\mathcal{G}_\mathrm{h}$ is due to the presence of wrong transformations with a better QLCP measure than the ground truth transformation. In this example, we clearly showed how the integration of texture features can relevantly improve the robustness of our registration algorithm, especially when they are extracted from rectified images. We recall that there are two ways of integrating texture features. The first one is to consider detected features only as sampled points, as for the experiments reported in Table 2. The second one (reported with the prefix "*match*") is to rank the matched features using their descriptors and keep in $\mathcal{P}$ the best k correspondences for each feature point. This experiment showed that: (1) the SURF features were more robust than the SIFT features in both $\mathcal{G}_\mathrm{g}$ and $\mathcal{G}_\mathrm{h}$; (2) the accuracy of the transformation obtained with SURF features is much better than the one obtained with SIFT features in $\mathcal{G}_\mathrm{g}$; (3) the features detected from the atlas of rectified reflectance images led to a more robust and generally accurate registration than those extracted from the original reflectance images (the improvement in term of accuracy is particularly evident in $\mathcal{G}_\mathrm{g}$, where the very low estimation error within 1-2 cm is 0.5 h of the bounding box diagonal length); (4) decreasing the number k of best feature correspondences based their descriptors greatly improves the efficiency (i.e., shorter computation times), but possibly deteriorates the robustness (i.e., lower successful rates); (5) in some experiments the features detected from the atlas of rectified reflectance images led to slightly higher estimation errors than those extracted from the original reflectance images. This happens when k is not large enough (e.g., k = 50, 100) and is mainly related to features located close to the rectified image boundaries. Those features extracted in these regions are less reliable due to the lack of enough texture information around them. This effect can be overcome by imposing a given degree of overlap between neighbouring rectified images.

Table 4 Performance evaluation of our algorithm using scene structure information

	$\mathcal{G}_\mathrm{d}$			$\mathcal{G}_\mathrm{f}$		
	Δ	***S**_r* **(%)**	***t* (s)**	**Δ**	***S**_r* **(%)**	***t* (s)**
Uniform	0.0498	100	42.2	0.2269	100	30.3
Uniform (scene structure)	0.0326	100	13.4	0.1500	100	10.5

As explained in Section 4.1, the proposed probability-based RANSAC is especially useful for aligning two scans with a small overlap. This advantage is evident in aligning the scan pair $\mathcal{G}_\mathrm{k}$ with a very small overlap (12%) as reported in Table 6, where S_r = 95% for the probabilistic RANSAC, but S_r = 65% for the basic RANSAC. The probability-based RANSAC led to similar results in both $\mathcal{G}_\mathrm{a}$ and $\mathcal{G}_\mathrm{e}$ with large overlaps, but with slightly better robustness in $\mathcal{G}_\mathrm{e}$. Note that K_p = 5, 000 and K_g = 100 for $\mathcal{G}_\mathrm{k}$ were used in this experiment.

Table 5 Performance evaluation on the texture feature-based variants of our algorithm with features detected on the original reflectance images and on atlases of rectified reflectance images

	$\mathcal{G}_\mathrm{g}$			$\mathcal{G}_\mathrm{h}$		
	Δ	***S*$_r$ (%)**	***t* (s)**	**Δ**	***S*$_r$ (%)**	***t* (s)**
Uniform	0.0727	100	32.2	★	0	53.1
SURF + uniform	0.0349	100	26.2	0.2632	60	11.0
matchSURF(*k* = 500) + uniform	0.0353	100	15.9	0.2555	55	4.6
matchSURF(*k* = 200) + uniform	0.0568	100	7.7	0.2532	40	4.6
matchSURF(*k* = 100) + uniform	0.0952	100	4.7	0.3667	50	4.4
matchSURF(*k* = 50) + uniform	0.1592	90	4.5	0.3201	25	3.1
Atlas: SURF + uniform	0.0148	100	17.8	0.2018	100	8.7
Atlas: matchSURF(*k* = 500) + uniform	0.0199	100	6.2	0.1560	85	6.1
Atlas: matchSURF(*k* = 200) + uniform	0.0248	100	5.1	0.2299	95	4.3
Atlas: matchSURF(*k* = 100) + uniform	0.0318	100	4.9	0.3095	75	4.0
Atlas: matchSURF(*k* = 50) + uniform	0.0367	100	2.5	0.3354	60	3.3
SIFT + uniform	0.0433	100	34.4	0.2351	30	11.5
matchSIFT(*k* = 500) + uniform	0.0548	100	15.3	0.2222	10	10.4
matchSIFT(*k* = 200) + uniform	0.0849	100	9.5	0.2801	10	3.9
matchSIFT(*k* = 100) + uniform	0.1674	100	5.9	★	0	3.4
matchSIFT(*k* = 50) + uniform	0.2842	85	5.5	★	0	1.5
Atlas: SIFT + uniform	0.0287	100	28.7	0.0494	30	10.8
Atlas: matchSIFT(*k* = 500) + uniform	0.0321	100	29.4	0.1584	25	4.3
Atlas: matchSIFT(*k* = 200) + uniform	0.0484	100	10.2	0.3768	15	5.3
Atlas: matchSIFT(*k* = 100) + uniform	0.1716	100	5.2	0.5138	10	4.4
Atlas: matchSIFT(k = 50) + uniform	0.3017	90	5.6	★	0	1.9

"★" denotes that no correct transformations 44 were found in 20 runs

Figure 7 shows seven outdoor scans acquired from the train station Gare de Lyon in Paris, which were automatically aligned by applying our NPCS algorithm followed by ICP registration. These seven scans were coupled by proximity to generate a set of scan pairs to register one after another. The generated pairs of scans have overlap rates of 39, 51, 31, 56, 25 and 29%. The first two scans in these seven scans comprise the tested scan pair $\mathcal{G}_\mathrm{j}$.

Furthermore, our algorithm was tested on two scans acquired from the same scene at different time instants with some changes as shown in Figure 8, where we acquired two scans almost at the same position after some people moved. In this example, the overlap rate is around 71%. Our algorithm always successfully and accurately registered these two scans (i.e., S_r = 100% and Δ = 2.86 cm).

Finally, we also compared our algorithm with the 4PCS algorithm [8] on five scan pairs $\mathcal{G}_\mathrm{a}$, $\mathcal{G}_\mathrm{c}$, $\mathcal{G}_\mathrm{i}$, $\mathcal{G}_\mathrm{j}$ and $\mathcal{G}_\mathrm{k}$. Initially we extracted 2,000 points from each scan using our proposed voxel-based uniform sampling approach. To make the comparison fair, we performed both the 4PCS algorithm and our proposed NPCS one with N = 4 using 600 randomly selected points from pre-extracted 2,000 points of each scan. Point surface normals (not available in $\mathcal{G}_\mathrm{a}$ and $\mathcal{G}_\mathrm{c}$) and same normal difference thresholds were used in both algorithms. The estimated overlapping rates of these five scan pairs shown in Figure 5 were used in 4PCS except for $\mathcal{G}_\mathrm{a}$ where 90% overlapping rate was used. However, in NPCS other parameters were fixed for all five tested scan pairs as mentioned before. Table 7 shows the performance comparison of both algorithms. The 4PCS

Table 6 Performance comparison between the basic RANSAC and the probabilistic RANSAC

Type of RANSAC	$\mathcal{G}_\mathrm{a}$			$\mathcal{G}_\mathrm{e}$			$\mathcal{G}_\mathrm{k}$ **(12% overlap)**		
	Δ	***S*$_r$ (%)**	***t* (s)**	**Δ**	***S*$_r$ (%)**	***t* (s)**	**Δ**	***Sr*(%)**	***t* (s)**
Basic RANSAC	0.8910	100	50.1	0.2160	95	25.3	0.9393	65	118.4
Probabilistic RANSAC	0.9013	100	55.7	0.2071	100	29.4	0.9543	95	102.7

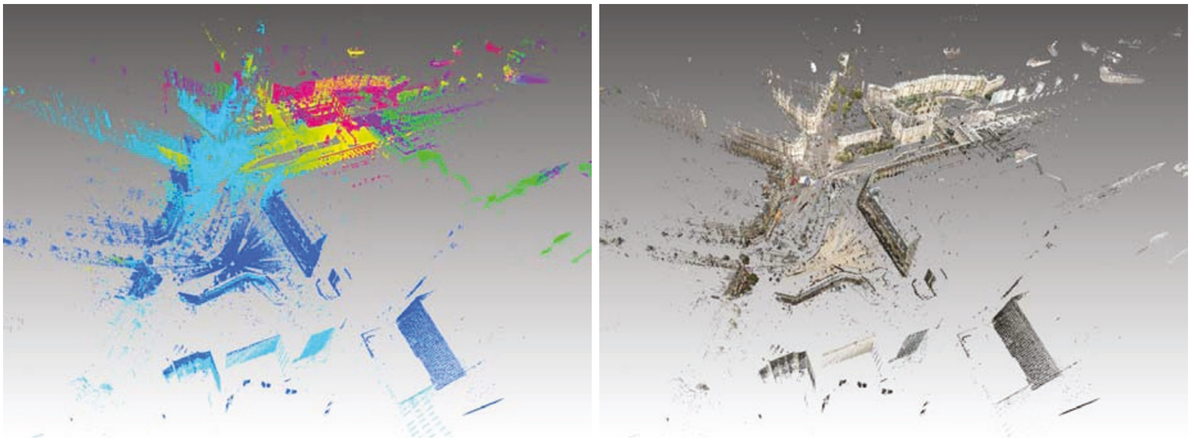


Figure 7 Automatically aligned point clouds of seven outdoor scans by applying our NPCS algorithm followed by ICP registration: (*left*) rendered by *colour* (each scan is represented in *different colour*); (*right*) rendered by *blending colour* images.

algorithm sometimes worked successfully in the first four tested scan pairs (S_r = 30-90%), but always failed in $\mathcal{G}_\mathrm{k}(S_r = 0\%)$. However, our NPCS algorithm almost always succeeded in the first four tested scan pairs and also succeeded in $\mathcal{G}_\mathrm{k}$ with a high successful rate (S_r = 60%). By utilizing the probability-based RANSAC, a higher successful rate can be obtained in $\mathcal{G}_\mathrm{k}$ (S_r = 95%, see Table 6). Note that the same maximal estimation errors for all tested scan pairs were used in the 4PCS algorithm for computing the estimation successful rate. However, the success of the 4PCS algorithm depends on the provided estimate for the overlap rate of two point sets to some extent. In our application, this overlap rate cannot be robustly approximated in advance. Thus, a feasible strategy would be to try it with different overlap rates until success. This obviously increases the overall execution time. However, in this comparison experiment, we directly used the pre-estimated overlapping rates with recovered ground truth transformations except that 90% overlapping rate was used in $\mathcal{G}_\mathrm{a}$. Another drawback of the 4PCS algorithm is related to the co-planarity constrain applied on the 4-points bases. Although this greatly reduces the search space, it may miss-lead the algorithm to false transformation estimations, especially in case of a very low overlap rate. We also noticed that in this case the computation time of the 4PCS algorithm increases significantly, since no successful transformations can be found efficiently. Finally,

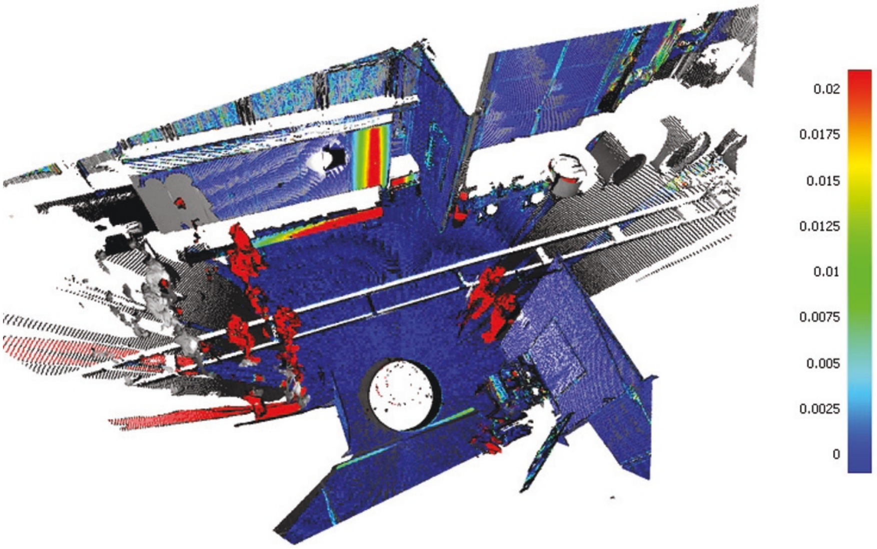


Figure 8 An example of the scene environment changed at two different acquisition time instants: the first scan is rendered with its reflectance and the second one is rendered with a colour map based on its changes w.r.t. the first one.

Table 7 Performance comparison of 4PCS and NPCS algorithms

	$\mathcal{G}_a$			$\mathcal{G}_c$			$\mathcal{G}_i$			$\mathcal{G}_j$			$\mathcal{G}_k$		
	Δ	S_r (%)	t (s)	Δ	S_r (%)	t (s)	Δ	S_r (%)	t (s)	Δ	S_r (%)	t (s)	Δ	S_r (%)	t (s)
4PCS	2.845	85	10.6	1.953	90	22.1	0.775	40	56.3	3.762	30	48.8	★	0	356.3
NPCS	0.875	100	46.3	0.420	100	31.2	0.458	95	45.5	0.517	100	13.5	0.980	60	76.2

the NPCS algorithm achieved a higher estimation accuracy than the 4PCS one especially in $\mathcal{G}_c$ and $\mathcal{G}_j$ mainly due to our proposed QLCP measure as reported in Table 3.

8 Conclusions

In this paper, we presented the NPCS registration, which is a robust and efficient approach for automatically aligning two point sets with overlap. Given two point sets $\mathcal{P}$ and $\mathcal{Q}$, our algorithm randomly searches for sets of congruent groups of N-points in $\mathcal{P}$ and $\mathcal{Q}$, which lead to the best estimate of the transformation. This is achieved by employing a RANSAC-based algorithm, which can also estimate the matching probability of each point to drive the search of possibly successful congruent bases of N-points. This probabilistic RANSAC approach improves the registration robustness especially for aligning two point sets with a small overlap. The search of congruent sets is efficiently performed by using a fast search codebook inspired by [20] to relevantly reduce the execution time. Our proposed search method can efficiently combine different metric functions to match points and point pairs in a multidimensional search space, which can be defined by geometric and texture feature descriptors and geometrical constraints of set of sampled points. This makes our framework general and flexible The efficient combination of feature descriptors can relevantly improve the registration performance, accuracy, and robustness of models with known specific characteristics. Moreover, we proposed a method to extract texture features from an atlas of rectified images recovered by sampling the reflectance image spherical field of view. The resulting feature descriptors were less sensitive to spherical distortions yielding more precise matches than those extracted from the original reflectance image.

To further improve the registration robustness, we introduced a new measure called QLCP. This is used to efficiently verify the rigid transformation estimated at each RANSAC iteration. The QLCP considers both the quantity and quality of the overlapping point set, and improved the verification robustness w.r.t. the LCP measure utilized in the 4PCS algorithm [8].

Our proposed NPCS algorithm was overall evaluated on a variety of input data with varying amount of noise, outliers, and extent of overlap (12-100%), acquired from small objects, indoor and outdoor scenes. The experiments focused on three main aspects: robustness (successful estimation rate), accuracy (median estimation error), and efficiency (median estimation time), respectively. In almost all cases, our NPCS algorithm (executed with the same parameters) successfully aligned two point clouds in less than a minute by accurately recovering their rigid transformations (the estimation errors are mostly within 0.5-5 h of their bounding box diagonal lengths). We tested our registration method with different sampling techniques and experimentally demonstrated the benefits of using methods generating uniformly distributed point samples w.r.t. random-based sampling strategies. Our proposed voxel-based uniform sampling approach was robust and efficient in almost all tested cases. We showed that the proposed QLCP measure is more reliable than the LCP measure. Still, it is possible to obtain wrong estimations, e.g., the failure related to the scan pair $\mathcal{G}_h$. Nevertheless, these wrong estimations can be successfully recovered by using texture-based features detected from the atlas of rectified reflectance images as shown by our experiments, or by using other suitable geometric features as those presented in [10,13,50]. In our experiments the NPCS registration performed, on average, better than the 4PCS method in terms of robustness and accuracy. In some experiments, in particular in presence of large overlaps, the execution time of our algorithm was slightly higher than 4PCS. While aligning two scans with a small overlap, our NPCS algorithm took similar computation time (mostly less than a minute) to achieve the good solution. However, 4PCS sometimes took more than 10 min, but still failed (e.g., in $\mathcal{G}_k$).

Future work will be focused on improving the performance of our method by exploiting a parallel hardware and to improve the robustness of our QLCP measure for inspection applications, which typically require to automatically align models captured at different times from scenes possibly containing changes.

Endnotes

[a] Only upper triangle part is stored in practice due to the symmetry of the point pairs lookup table. [b] The demo application of 4PCS algorithm [8] is available at http://graphics.stanford.edu/~niloy/research/fpcs/4PCS_demo.html.

Algorithm 1

Find the best K_g N-points bases $\{\mathcal{B}_q^k\}_{k=1}^{K_g}$ in $\mathcal{Q}$ which are approximate congruent to the N-points base $\mathcal{B}_p$ in $\mathcal{P}$.

```
$\mathcal{M} \leftarrow \{(\mathbf{q}_1^k, \mathbf{q}_2^k)\}_{k=1}^{K_p}$, i.e., the best $K_p$ point pairs in $\mathcal{Q}$ possibly matching the point pair $(\mathbf{p}_1, \mathbf{p}_2)$ in $\mathcal{B}_p$, according with the similarity score in Equation 3.
for $i$ = 2 to $N$ - 1 do
    case 1: $\mathcal{S} \leftarrow \{(\mathbf{q}_i^k, \mathbf{q}_{i+1}^k)\}_{k=1}^{K_p}$ in $\mathcal{Q}$ possibly matching the point pair $(\mathbf{p}_i, \mathbf{p}_{i+1})$ $\mathcal{B}_p$.
    case 2: collect $\mathcal{M}_f(\mathbf{p}_{i+1})$
    $\mathcal{M}' \leftarrow \emptyset$
        for each group $(\mathbf{q}_1^m, \ldots, \mathbf{q}_i^m)$ of points in $\mathcal{M} = \{(\mathbf{q}_1^m, \ldots, \mathbf{q}_i^m)\}_{m=1}^{|\mathcal{M}|}$do
            case 1: group point pair
                for each point pair $(\mathbf{q}_i^k, \mathbf{q}_{i+1}^k)$ in $\mathcal{S}$do
                    if $\mathbf{q}_i^m = \mathbf{q}_i^k$ and $S_f(\mathbf{p}_{i+1}, \mathbf{q}_{i+1}^k) \prod_{j=1}^{i-1} \prod_{k=1}^{K} s_k(m_k(\mathbf{p}_j^m, \mathbf{p}_{i+1}),\ m_k(\mathbf{q}_j, \mathbf{q}_s)) = 1$ then
                        $\mathcal{M}' \leftarrow \{\mathcal{M}', (\mathbf{q}_1^m, \ldots, \mathbf{q}_i^m,\ \mathbf{q}_{i+1}^k)\}$
                    end if
                end for
            case 2: add a single point
                for each point $\mathbf{q}_s \in \mathcal{M}_f(\mathbf{p}_{i+1})$do
                    if $\prod_{j=1}^{i} \prod_{k=1}^{K} s_k(m_k(\mathbf{p}_j^m, \mathbf{p}_{i+1}),\ m_k(\mathbf{q}_j, \mathbf{q}_s)) = 1$ then
                        $\mathcal{M}' \leftarrow \{\mathcal{M}', (\mathbf{q}_1^m, \ldots, \mathbf{q}_i^m,\ \mathbf{q}_s)\}$
                    end if
                end for
    end for
    $\mathcal{M} \leftarrow \mathcal{M}'$
end for
return $\{\mathcal{B}_q^k\}_{k=1}^{K_g}$ by filtering $\mathcal{M}$ according with Eq. (3)
```

Competing interests
The authors declare that they have no competing interests.

Received: 26 May 2011 Accepted: 23 September 2011
Published: 23 September 2011

References

1. RI Hartley, A Zisserman, *Multiple View Geometry in Computer Vision*, 2nd edn. (Cambridge University Press, 2004). ISBN: 0521540518
2. J Yao, WK Cham, Robust multi-view feature matching from multiple unordered views. Pattern Recogn. **40**(11), 3081–3099 (2007). doi:10.1016/j.patcog.2007.02.011
3. M Pollefeys, L Van Gool, M Vergauwen, F Verbiest, K Cornelis, J Tops, R Koch, Visual modeling with a hand-Held camera. Int J Comput Vis. **59**(3), 207–232 (2004)
4. W Zhang, J Yao, WK Cham, 3D modeling from multiple images, in *The Seventh International Symposium on Neural Networks (ISNN 2010)*, 97–103 (2010)
5. PJ Besl, HD McKay, A method for registration of 3-D shapes. IEEE Trans Pattern Anal Mach Intell. **14**(2), 239–256 (1992). doi:10.1109/34.121791
6. S Rusinkiewicz, M Levoy, Efficient Variants of the ICP Algorithm, in *International Conference on 3-D Digital Imaging and Modeling (3DIM)* (2001)
7. C Matabosch, J Salvi, D Fofi, F Meriaudeau, Range image registration for industrial inspection, in *Mach Vis Appl Ind Insp XIII*, 216–227 (2005)
8. D Aiger, NJ Mitra, D Cohen-Or, 4-Points Congruent Sets for Robust Pairwise Surface Registration, in *ACM SIGGRAPH* (2008)
9. MA Fischler, RC Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM. **24**(6), 726–740 (1987)
10. N Gelfand, NJ Mitra, LJ Guibas, H Pottmann, Robust Global Registration, in *Third Eurographics Symposium on Geometry Processing (SGP)* (2005)
11. L Yu, D Zhang, E Holden, A fast and fully automatic registration approach based on point features for multi-source remote-sensing images. Computers Geosci. **34**(7), 838–848 (2008). doi:10.1016/j.cageo.2007.10.005
12. Z Kang, Automatic Registration of Terrestrial Point Cloud Using Panoramic Reflectance Images, in *International Society for Photogrammetry and Remote Sensing* (2008)
13. A Zaharescu, E Boyer, K Varanasi, R Horaud, Surface Feature Detection and Description with Applications to Mesh Matching, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009)
14. JS Chalfant, NM Patrikalakis, Three-dimensional object registration using wavelet features. Eng Computers. **25**(3), 303–318 (2009). doi:10.1007/s00366-009-0126-5
15. ER Smith, RJ Radke, CV Stewart, Physical Scale Intensity-Based Range Keypoints, in *3D Data Processing, Visualization, and Transmission (3DPVT)* (2010)
16. AE Johnson, M Hebert, Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans Pattern Anal Mach Intell. **21**(5), 433–449 (1999). doi:10.1109/34.765655
17. DF Huber, M Hebert, Fully automatic registration of multiple 3D data sets. Image Vis Comput. **21**(7), 637–650 (2003). doi:10.1016/S0262-8856(03)00060-X
18. A Makadia, AI Patterson, K Daniilidis, Fully Automatic Registration of 3D Point Clouds, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2006)
19. I Stamos, M Leordeanu, Automated Feature-based Range Registration of Urban Scenes of Large Scale, in *IEEE Conference on Computer Vision and Pattern Recognition* (2003)
20. J Yao, MR Ruggeri, P Taddei, V Sequeira, Automatic scan registration using 3D linear and planar features. 3D Res. **1**(3), 1–18 (2010)
21. C Chao, I Stamos, Semi-automatic Range to Range Registration: a Feature-based Method, in *International Conference on 3-D Digital Imaging and Modeling (3DIM)* (2005)
22. C Dold, C Brenner, Registration of Terrestrial Laser Scanning Data Using Planar Patches and Image Data, in *International Society for Photogrammetry and Remote Sensing* (2006)
23. C Chao, I Stamos, Range Image Registration Based on Circular Features, in *Proceedings of International Symposium on 3D Data Processing Visualization and Transmission (3DPVT)* (2006)
24. M Franaszek, GS Cheok, C Witzgall, Fast automatic registration of range images from 3D imaging systems using sphere targets. Autom Constr. **18**(3), 265–274 (2009). doi:10.1016/j.autcon.2008.08.003
25. T Rabbani, F van den Heuvel, Automatic Point Cloud Registration Using Constrained Search for Corresponding Objects, in *7th Conference on Optical 3-D Measurement Techniques* (2005)
26. L Silva, OR Bellon, KL Boyer, Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. IEEE Trans Pattern Anal Mach Intell. **27**, 762–776 (2005)
27. F Boughorbel, M Mercimek, A Koschan, MA Abidi, A new method for the registration of three-dimensional point-sets: the Gaussian fields framework. Image Vis Comput. **28**(1), 124–137 (2010). doi:10.1016/j.imavis.2009.05.003
28. S Gold, A Rangarajan, CP Lu, S Pappu, E Mjolsness, New algorithms for 2D and 3D point matchingpose estimation and correspondence. Pattern Recogn. **31**(8), 1019–1031 (1998). doi:10.1016/S0031-3203(98)80010-1
29. S Granger, X Pennec, Multi-scale EM-ICP: A Fast and Robust Approach for Surface Registration, in *Proc. of the 7th European Conference on Computer Vision-Part IV*, ECCV '02, London, UK, UK (Springer-Verlag, 2002), pp. 418–432
30. Y Liu, Automatic range image registration in the Markov chain. IEEE Trans Pattern Anal Mach Intell. **32**(1), 12–29 (2010)
31. T Tamaki, M Abe, B Raytchev, K Kaneda, Softassign and EM-ICP on GPU, in *International Conference on Networking and Computing (ICNC)*, 179–183 (2010)

32. L Juan, O Gwon, A Comparison of SIFT, PCA-SIFT and SURF. Int J Image Process IJIP. **3**(4), 143–152 (2009)
33. S Arya, DM Mount, NS Netanyahu, R Silverman, AY Wu, An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions, in *ACM-SIAM Symposium on discrete algorithms* (1994)
34. Z Zhang, OD Faugeras, Determining motion from 3D line segment matches: a comparative study. Image Vis Comput. **9**, 10–19 (1991). doi:10.1016/0262-8856(91)90043-O
35. N Gelfand, L Ikemoto, S Rusinkiewicz, M Levoy, Geometrically Stable Sampling for the ICP Algorithm, in *Fourth International Conference on 3D Digital Imaging and Modeling (3DIM)* (2003)
36. BJ Brown, S Rusinkiewicz, Global non-rigid alignment of 3-D scans. ACM Trans Graph. **26**(3), 21 (2007). doi:10.1145/1276377.1276404
37. A Torsello, E Rodola, A Albarelli, Sampling Relevant Points for Surface Registration, in *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT2011)* (2011)
38. D Nehab, P Shilane, Stratified Point Sampling of 3D Models, in *Proc of the Symposium on Point-Based Graphics*, 49–56 (2004)
39. MR Ruggeri, G Patanè, M Spagnuolo, D Saupe, Spectral-driven isometry-invariant matching of 3D shapes. Int J Comput Vis. **89**, 248–265 (2010). doi:10.1007/s11263-009-0250-0
40. J Bowers, R Wang, LY Wei, D Maletz, Parallel Poisson disk sampling with spectrum analysis on surfaces. ACM Trans Graph. **29**, 166:1–166:10 (2010)
41. M Pauly, M Gross, LP Kobbelt, Efficient simplification of point-sampled surfaces, in *Proc of Vis*, 163–170 (2002)
42. J Yao, P Taddei, MR Ruggeri, V Sequeira, Complex and photo-realistic scene representation based on range planar segmentation and model fusion. Int J Robotics Res. **30**(10), 1263–1283 (2011). doi:10.1177/0278364911410754
43. T Tuytelaars, K Mikolajczyk, Local invariant feature detectors: a survey. Found Trends Comput Graph Vis. **3**(3), 177–280 (2008)
44. DG Lowe, Distinctive image features from scale-invariant keypoints. Int J Computer Vis. **60**(2), 91–110 (2004)
45. H Bay, A Ess, T Tuytelaars, L Van Gool, Speeded-up robust features (SURF). Comput Vis Image Underst. **110**(3), 346–359 (2008). doi:10.1016/j.cviu.2007.09.014
46. E Rosten, T Drummond, Machine Learning for High-speed Corner Detection, in *European Conference on Computer Vision*. **1**, 430–443 (2006)
47. SN Sinha, J michael Frahm, M Pollefeys, Y Genc, GPU-based Video Feature Tracking and Matching. Tech. rep., in Workshop on Edge Computing Using New Commodity Architectures (2006)
48. E Rosten, R Porter, T Drummond, Faster and better: a machine learning approach to corner detection. IEEE Trans Pattern Anal Mach Intell. **32**(1), 105–119 (2010)
49. C Harris, M Stephens, A Combined Corner and Edge Detector, in *The Fourth Alvey Vision Conference*, 147–151 (1988)
50. A Albarelli, E Rodola, A Torsello, Loosely Distinctive Features for Robust Surface Alignment, in *European Conference on Computer Vision (ECCV2010)*, 519–532 (2010)

doi:10.1186/1687-6180-2011-72
Cite this article as: Yao *et al.*: **Robust surface registration using *N*-points approximate congruent sets.** *EURASIP Journal on Advances in Signal Processing* 2011 **2011**:72.