

Chapter 8

3D Shape Registration



Umberto Castellani and Adrien Bartoli

Abstract Registration is the problem of bringing together two or more 3D shapes, either of the same object or of two different but similar objects. This chapter first introduces the classical Iterative Closest Point (ICP) algorithm which represents the gold standard registration method. Current limitations of ICP are addressed, and the most popular variants of ICP are described to improve the basic implementation in several ways. Challenging registration scenarios are analyzed, and a taxonomy of recent and promising alternative registration techniques is introduced. Four case studies are then described with an increasing level of difficulty. The first case study describes a simple but effective technique to detect outliers. The second case study uses the Levenberg–Marquardt (LM) optimization procedure to solve standard pairwise registration. The third case study focuses on the challenging problem of deformable object registration. The fourth case study introduces an ICP method for preoperative data registration in laparoscopy. Finally, open issues and directions for future work are discussed, and conclusions are drawn.

8.1 Introduction

Registration is a critical issue for various problems in computer vision and computer graphics. The overall aim is to find the best alignment between two objects or between several instances of the same object, in order to bring the shape data into the same reference system. The main high-level problems that use registration techniques are as follows:

1. **Model reconstruction.** The goal in model reconstruction is to create a complete object model from partial 3D views obtained by a 3D scanner. Indeed, it is rare that

U. Castellani (✉)
University of Verona, Verona, Italy
e-mail: Umberto.Castellani@univr.it

A. Bartoli
Université d’Auvergne, Clermont-Ferrand, France
e-mail: Adrien.Bartoli@gmail.com

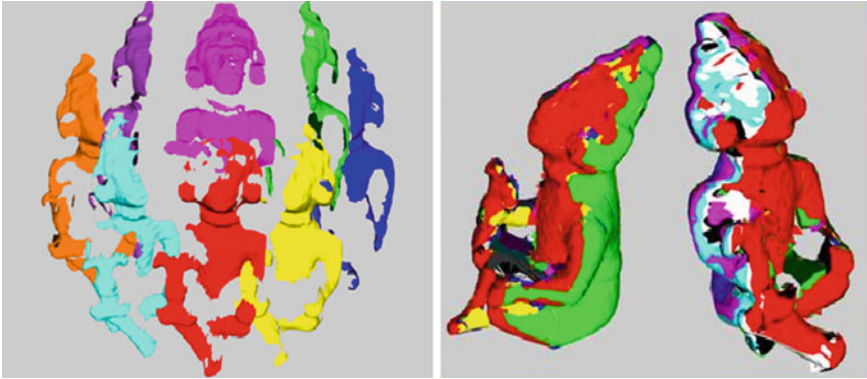


Fig. 8.1 Example of model reconstruction. Partial 3D views of the object of interest are acquired (left). After registration, all the 3D views are transformed to the common reference system and merged (right)

a single 3D view grabs the whole object structure, mainly due to self-occlusions. Registration allows one to obtain the alignment between the partial overlapping 3D views in order to build a complete object model, also called a mosaic (see Fig. 8.1). In this context, registration is first applied between pairs of views [11, 126]. The whole model is then reconstructed using multiple-view registration refinement [72, 126]. Typically, model reconstruction is employed in *cultural heritage* [10] to obtain 3D models of archeological findings. It has also been applied in applications such as *reverse engineering* and *rapid prototyping* [150] and for vision in hostile environments [29, 30].

2. **Model fitting.** The goal in model fitting is to compute the transformation between a partial 3D view and a known CAD model of the actual object. Model fitting is used in robotics for object grasping [41, 109] and model-based object tracking [123]. Model fitting is typically used with rigid objects but has recently been extended to deformable objects [31].
3. **Object recognition.** The goal in object recognition is to find, among a database of 3D models, which one best matches an input partial 3D view. This problem is more challenging than model fitting since a decision has to be made regarding which model, if any, is the sought one. Solving the recognition problem this way is called *recognition-by-fitting* [147]. Several works have been done for 3D face recognition [18, 20, 133] and for 3D object retrieval [56, 143]. Registration becomes more challenging in a cluttered environment [8, 76, 95].
4. **Multimodal registration.** The goal in multimodal registration is to align several views of the same object taken by different types of acquisition systems. After registration, the information from different modalities can be merged for comparison purposes or for creating a multimodal object model. This problem is typical in medical imaging where it is CT scans or MRI and PET scans [89, 134]. 3D medical image registration is discussed further in Chap. 11.

This chapter gives a general formulation of the registration problem. This formulation leads to computational solutions that can be used to solve the four above-mentioned tasks. It encompasses most of the existing registration algorithms. For a detailed description of registration techniques and experimental comparisons, we refer the reader to recent surveys [77, 96, 124, 126, 128, 141, 142]. It is worth mentioning that most of the existing computational solutions are based on the seminal Iterative Closest Point (ICP) [11] algorithm that we will describe shortly.

8.1.1 Chapter Outline

This chapter is organized as follows. We first present the two-view registration problem and the current algorithmic solutions. We then describe some advanced registration techniques. We give a comprehensive derivation of algorithms for registration by proposing four case studies. We give an overview of open challenges with future directions and conclusion. Further suggestions, additional reading, and exercises are finally proposed.

8.2 Registration of Two Views

We first give a mathematical formulation of the two-view registration problems and then derive the basic ICP algorithm and discuss its main variants.

8.2.1 Problem Statement

Given a pair of views \mathbb{D} and \mathbb{M} representing two scans (partial 3D views) of the same object, registration is the problem of finding the parameters \mathbf{a} of the transformation function $T(\mathbf{a}, D)$ which best aligns D to M . Typically, *set* D and \mathbb{M} are either simple point clouds or triangulated meshes [26]. The moving view \mathbb{D} is called *data-view*, while the fixed view \mathbb{M} is called *model-view*. The registration problem is solved by estimating the parameters \mathbf{a}^* of the transformation T that satisfy

$$\mathbf{a}^* = \arg \min_{\mathbf{a}} E(T(\mathbf{a}, \mathbb{D}), \mathbb{M}), \quad (8.1)$$

where E is called the *error function* and measures the registration error. Figure 8.2 illustrates the two-view registration process. The data-view and the model-view show different portions of *Bunny*. The transformation function $T(\mathbf{a}, \mathbb{D})$ is applied, and the registered views are shown.

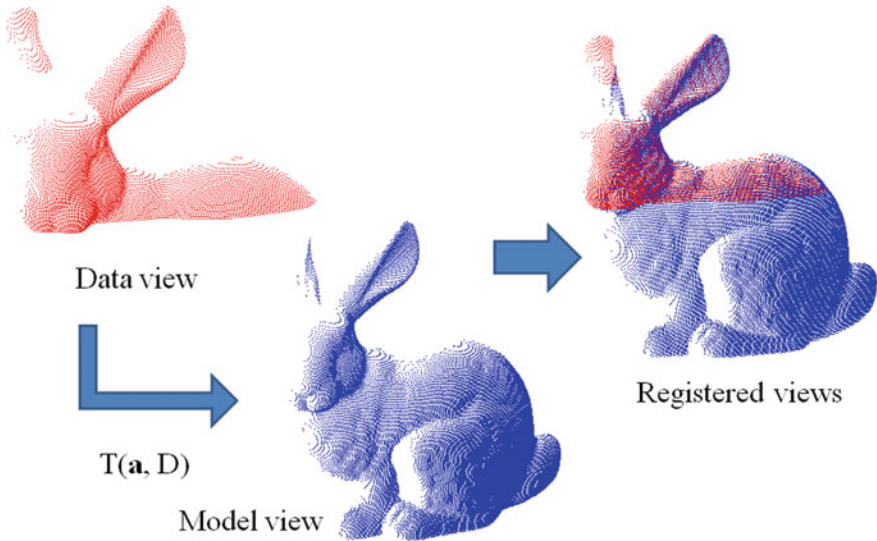


Fig. 8.2 Pairwise registration. The data-view and the model-view (left) are registered. The transformation function $T(\mathbf{a}, \mathbb{D})$ allows one to move the data-view to the model-view coordinate frame (right)

Most of the registration methods are based on the paradigm defined directly above and differ in the following aspects:

- **The transformation function.** The transformation function T usually implements a rigid transformation of the 3D space. It uses a translation vector \mathbf{t} and a rotation matrix \mathbf{R} whose values are encoded or parametrized in the parameter vector \mathbf{a} . The transformation function may also handle deformations; this requires a more complex formulation.
- **The error function.** The error function E measures the registration error or dissimilarity between \mathbb{D} and \mathbb{M} after alignment. When the transformation function T is rigid, E is a measure of *congruence* between the two views. In general, E takes the form of an L_2 approximation of the Hausdorff distance which further involves the so-called *point-to-point* distance [11] or the *point-to-plane* distance [36].
- **The optimization method.** This is the method or algorithm used to find the minimizer \mathbf{a} in problem (8.1). The gold standard is the ICP algorithm [11] which was specifically designed for the problem at hand. General-purpose optimization methods such as Levenberg–Marquardt [54] have also been used for this problem.

8.2.2 The Iterative Closest Points (ICP) Algorithm

In the classical ICP algorithm [11], the overall aim is to estimate a rigid transformation with parameters $\mathbf{a}^* = (\mathbf{R}, \mathbf{t})$. Both views are treated as point clouds $\mathbb{D} = \{\mathbf{d}_1, \dots, \mathbf{d}_{N_d}\}$ and $\mathbb{M} = \{\mathbf{m}_1, \dots, \mathbf{m}_{N_m}\}$. The error function is chosen as

$$E_{ICP}(\mathbf{a}, \mathbb{D}, \mathbb{M}) = \sum_{i=1}^{N_d} \|(\mathbf{R}\mathbf{d}_i + \mathbf{t}) - \mathbf{m}_j\|^2, \quad (8.2)$$

where we define $E_{ICP}(\mathbf{a}, \mathbb{D}, \mathbb{M}) = E(T(\mathbf{a}, \mathbb{D}), \mathbb{M})$ and where $(\mathbf{d}_i, \mathbf{m}_j)$ are corresponding points [126].¹ Fixing $\mathbf{d}_i \in \mathbb{D}$ the corresponding point $\mathbf{m}_j \in \mathbb{M}$ is computed such that

$$j = \arg \min_{k \in \{1, \dots, N_m\}} \|(\mathbf{R}\mathbf{d}_i + \mathbf{t}) - \mathbf{m}_k\|^2. \quad (8.3)$$

More specifically, the value

$$e_i^2 = \|(\mathbf{R}\mathbf{d}_i + \mathbf{t}) - \mathbf{m}_j\|^2 \quad (8.4)$$

is the square of the *residual*. Figure 8.3 illustrates the step of correspondence computation. For each data point (in red), the closest model point (in blue) is computed using the Euclidean distance. The list of correspondences is thus obtained. Note that, given point correspondences, computation of \mathbf{R} and \mathbf{t} to minimize E_{ICP} in Eq. 8.2 can be solved in closed form [126]. Several approaches are possible for the closed-form, least squares estimation of this 3D rigid body transformation. These include approaches based on Singular Value Decomposition (SVD), unit quaternion, dual quaternion, and orthonormal matrices. Although the study of Eggert et al. [49] found little difference in the accuracy and robustness of all these approaches, perhaps the most well known of these is the SVD approach by Arun et al. [5]. Here, the cross-covariance matrix is formed for the N_d correspondences, $(\mathbf{d}_i, \mathbf{m}_j)$, as

$$\mathbf{C} = \frac{1}{N_d} \sum_{i=1}^{N_d} (\mathbf{d}_i - \bar{\mathbf{d}})(\mathbf{m}_j - \bar{\mathbf{m}})^T, \quad (8.5)$$

where the means $\bar{\mathbf{d}}, \bar{\mathbf{m}}$ are formed over the N_d correspondences. Performing the SVD of \mathbf{C} gives us

$$\mathbf{U}\mathbf{S}\mathbf{V}^T = \mathbf{C}, \quad (8.6)$$

where \mathbf{U} and \mathbf{V} are two orthogonal matrices and \mathbf{S} is a diagonal matrix of singular values. The rotation matrix \mathbf{R} can be calculated from the orthogonal matrices as

¹Note that the pair $(\mathbf{d}_i, \mathbf{m}_j)$ is initially a putative correspondence, which becomes a true correspondence when convergence to a global minimum is attained.

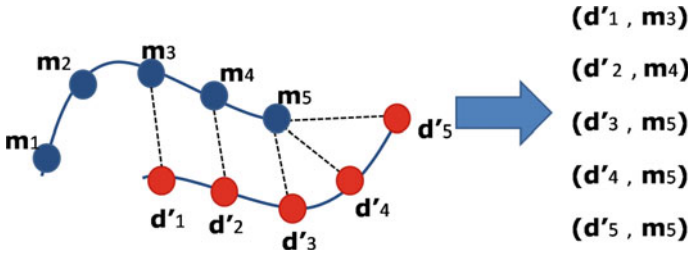


Fig. 8.3 Correspondence estimation in ICP. For each transformed data point $\mathbf{d}'_i = \mathbf{R}\mathbf{d}_i + \mathbf{t}$, the closest model point \mathbf{m}_j is estimated (left). The list of corresponding points is then defined (right)

$$\mathbf{R} = \mathbf{V}\mathbf{U}^T \quad (8.7)$$

This solution may fail to give a correct rotation matrix and give a reflection instead when the data is severely corrupted [149]. Thus, it can be modified to always return a correct rotation matrix [149]:

$$\mathbf{R} = \mathbf{V}\mathbf{S}\mathbf{U}^T \quad (8.8)$$

where

$$\mathbf{S} = \begin{cases} \mathbf{I} & \text{if } \det(\mathbf{U})\det(\mathbf{V}) = 1 \\ \text{Diag}(1, 1, \dots, 1, -1) & \text{if } \det(\mathbf{U})\det(\mathbf{V}) = -1. \end{cases}$$

Once the rotation matrix has been estimated, the translation vector \mathbf{t} can be estimated as

$$\mathbf{t} = \bar{\mathbf{m}} - \mathbf{R}\bar{\mathbf{d}} \quad (8.9)$$

The ICP algorithm is iterative because it iteratively improves the putative correspondences. If true correspondences were known, clearly the process could operate in one shot (one pass). ICP has two main steps in its inner loop: (i) closest point computation and (ii) rigid transformation estimation. In more detail, the algorithm operates as follows:

1. For each data point $\mathbf{d}_i \in \mathbb{D}$, compute the closest point $\mathbf{m}_j \in \mathbb{M}$ according to Eq. 8.3.
2. With the correspondences $(\mathbf{d}_i, \mathbf{m}_j)$ from step 1, estimate the new transformation parameters $\mathbf{a} = (\mathbf{R}, \mathbf{t})$.
3. Apply the new transformation parameters \mathbf{a} from step 2 to the point cloud \mathbb{D} .
4. If the change in $E_{ICP}(\mathbf{a}, \mathbb{D}, \mathbb{M})$ between two successive iterations is lower than a threshold then terminate, else go to step 1.

It was proven [11] that this algorithm is guaranteed to converge monotonically to a local minimum of Eq. (8.2). Note that, as for any local iterative method, a strategy for initializing \mathbf{a} must be used. An overview of the most popular initialization strategies is given in Sect. 8.2.3.1.

8.2.3 ICP Extensions

Although ICP has been successfully applied to many registration problems, there are several critical issues that need to be taken care of. In particular, ICP performs well when the following assumptions are met:

1. *The two views must be close to each other.* If not, ICP will probably get stuck in a local minimum. This issue is typically solved by pre-alignment of the two 3D views, also called *coarse* registration.
2. *The two views must fully overlap or the data-view \mathbb{D} must be a subset of the model-view \mathbb{M} .* The problem arises from the fact that ICP always assigns a closest point to every data point. If a data point has no corresponding model point, this will create a spurious correspondence, an *outlier* with respect to the sought transformation, that will bias the solution or prevent the algorithm from finding the correct transformation parameters.

Two other important issues are the *speed* of computation and the *accuracy* of the ICP algorithm. Typically, methods focused on speed improvement for the closest point computation step which is the bottleneck of the algorithm. Other interesting approaches address instead of the speed of convergence by proposing new distance formulations for problem (8.1). Methods focusing on accuracy exploit additional information in order to measure the *similarity* between corresponding points not only in terms of proximity. In the following, we describe some registration techniques which improve the basic ICP method in several ways. Figure 8.4 illustrates the proposed taxonomy of ICP extensions so as to easily understand the organization of previous work in this field.

8.2.3.1 Techniques for Pre-alignment

The aim of pre-alignment techniques is to estimate a coarse transformation which will allow the two views to get closer. This helps the data-view to be transformed within basin of attraction of the correct local minimum. In practice, instead of searching *dense* point-to-point correspondences, pre-alignment techniques estimate the best matching between *features* extracted from the views. Roughly speaking, the features can be *global* or *local*. The former is a compact representation that effectively and concisely describes the entire view. The latter instead is a collection of local and discriminative descriptors computed on sub-parts of the views.

Global Approaches

Global approaches typically estimate and match the principal coordinate system of each view. The simplest approach is to compute the main translational alignment by shifting the centroids of the two point clouds to the origin of the coordinate system (i.e., zero-mean). In order to estimate also the orientation of the principal axes, Principal Component Analysis (PCA) to the 3D points can be performed. The problems with PCA as a pre-alignment method are (i) a 180° ambiguity in the

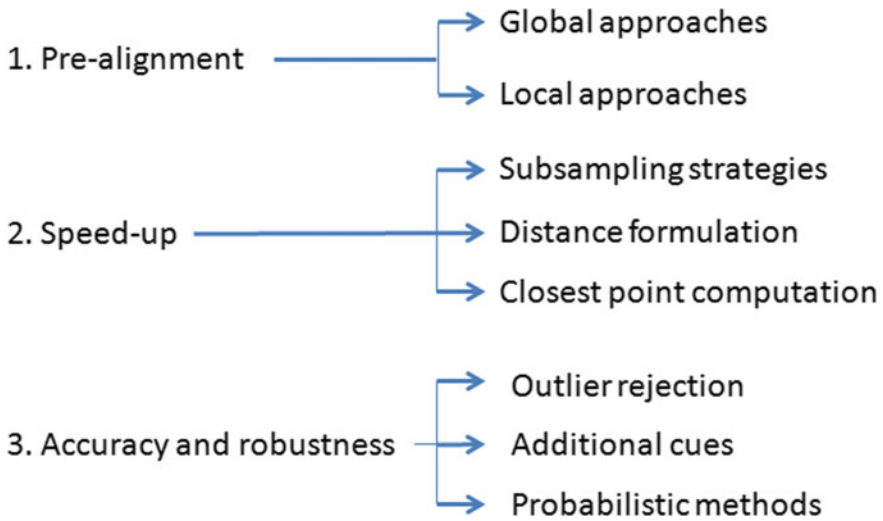


Fig. 8.4 A taxonomy of some ICP extensions

direction of the principal axes, (ii) principal axes may switch for shapes that have eigenvalues similar in value, particularly if the object is able to deform slightly, and (iii) a vulnerability to outliers in the raw shape data (as discussed). Even if we enforce a right-handed frame using the sign of cross-product of basis vectors, there still exists an overall 180° ambiguity, unless higher order moments are used. Moments of higher orders are also useful to improve accuracy [21]. Of course, these approaches perform well when the two views fully overlap. Otherwise, the non-overlapping parts change the estimation of the principal axes and thus affect the pre-alignment. Some improvements have been made by extracting and matching the *skeletons* of the views [32, 99] but this is feasible for articulated objects only. Recently, a method for registration between views in arbitrary pose was proposed as the so-called *GO-ICP* method [157]. The key idea consists of solving the optimization problem using a branch and bound algorithm to guarantee the estimation of a global solution independently of the initialization.

Local Approaches

Local approaches define a *descriptor* (or signature) for each 3D point which encodes local shape variation in the point neighborhood [27, 76, 78, 104, 140]. See also [63, 64, 145] for a comprehensive survey on local geometric descriptors. Point correspondences are then obtained as the best matches in regard to the point signatures. Various methods to compute signatures were proposed. In the seminal work [76], the *spin images* were introduced. In a spin image, the neighbors of some selected 3D point (e.g., a 3D interest point [145]) are binned in a 2D cylindrical-polar coordinate system. This consists of a distance from the selected point within that point's tangent plane and a signed height above/below the tangent plane. Thus, the spin image is a 2D histogram of 3D shape, where one dimension of information is sacrificed for

pose invariance. In [80], curvilinear features on the object are estimated from a small amount of points of interest. Gaussian and mean curvatures are used to this aim. Similarly, in [156], bitangent curve pairs were used as landmarks on the surface. In [104], a geometric scale-space analysis of 3D models was proposed from which a scale-dependent local shape descriptor was derived. Similarly in [27], registration involves few feature points by extending the approach for salient point detection to the 3D domain. A *generative model* is then estimated as a point descriptor by using hidden Markov models. In [78], the proposed descriptor encodes not only local information around the point, but also inter-point relationships. The method is inspired by the so-called *Shape Context* [9] which was improved using the *Bag-of-Words* paradigm [42]. Note that from the analysis of inter-point relationships, it is also possible to estimate the overlapping region between two views. It is worth noting that in general the estimation of the overlap area is not trivial. An interesting approach was proposed in [131] by combining local geometric features with advanced graph matching techniques. The method consists of representing all putative point matches as a graph, and then selecting as many *consistent* matches among them as possible. To this aim, a global discrete optimization problem is proposed based on the so-called *maximum strict sub-kernel algorithm* [130].

8.2.3.2 Techniques for Improving Speed

The speed of the algorithm is crucial for many applications. Unfortunately, when the number of points is very high, the basic ICP algorithm becomes very slow. In order to address this issue, several strategies were proposed. Many of these strategies are implemented by the Kinect fusion toolkit [101] for real-time modeling using dynamic RGBD sensors.

Subsampling

Subsampling can be applied to either the data-view only or to both the data-view and the model-view. *Random* and *uniform* strategies are common approaches [126]. *Normal space* sampling is a more sophisticated approach based on choosing points such that the distribution of normals among the selected points is as spread as possible. This increases the influence of smaller details which are crucial to better disambiguate the rigid transformation due to translational sliding. Another effective practice is based on a *hierarchical* subsampling where the *range* map is re-organized in a pyramidal fashion to obtain a coarse to fine representation of the source data. This approach is employed for online modeling as in [101] where the coarse levels with few points are used to estimate the rough motion, and vice-versa the detailed levels that involve more points are employed for refining the alignment.

Closest Point Computation

As mentioned above, closest point computation is the bottleneck of the registration process due to the quadratic complexity ($O(n^2)$) in finding the correspondence of each point. Early strategies were based on the organization of the model points in a k - d tree [136] structure in order to reduce the closest point complexity to $O(n \log n)$.

Closest point caching [136] also accelerates the speed of ICP (the data point correspondence search is only among a subset of model points which were the closest at the previous iteration). Indeed, in [105] k - d tree and caching are combined in order to further improve the speed of ICP. Other more effective approaches are based on the so-called *reverse calibration* paradigm [13]. The idea is to project the source data point onto the destination model-view which is encoded as a *range* image [124]. In particular, the projection from the 3D domain into the range image is performed by using the calibration parameters of the 3D scanner. In this fashion, the correspondence is computed in one shot. The reverse calibration approach is especially effective for real-time modeling purposes [30, 101, 125]. For instance, in [125] the authors proposed the first real-time 3D model reconstruction system where the full modeling pipeline is carried out during the acquisition using a real-time sensor. In [30], online registration is performed to build a 3D mosaic of the scene in order to improve the navigation in underwater environments. This online modeling using reverse calibration alignment has been consolidated more recently with the large availability of dynamic RGBD sensors [101]. Note that the one-shot computation can be carried out also on generic point cloud (not necessarily coming from a range image) by precomputing the so-called *distance transform* of the model-view [54]. Figure 8.5 illustrates the distance transform. In practice, the distance to closest model points is pre-computed for all grid points of the discretized volume. The case for distance transform computed for the model is particularly compelling when one wishes to align many instances of data scan to the same model scan. A more recent class of methods is based on the GPU implementation of data representation to improve the computation of corresponding points [47, 48]. A probabilistic approach is proposed using Gaussian Mixture Models (GMMs) where a decoupling technique is introduced for parallel estimation of parameters. Also, in [101], a parallel computation of local contributions of distance estimation is employed on a GPU to improve the speed of real-time modeling.

Distance Formulation

Another crucial factor affecting the speed of ICP is the point-to-point or point-to-plane distance used in problem (8.1). Figure 8.6 shows a schema of the two kinds of distances: point-to-point computes the euclidean distance between the data point and model point (left), and point-to-plane distance computes the projection of the data point onto the surface of the model-view which is encoded in terms of piecewise planar patches (for instance, a triangular mesh). In spite of an increased complexity of the distance for point-to-plane formulation, the number of ICP iterations required to converge is reduced [110, 115]. Whether this results in a reduced registration time depends on the tradeoff between the increased per-iteration time and the reduced number of iterations. Note that usually the point-to-plane distance is employed for real-time modeling pipelines [101, 124] where pairwise registration is carried out for subsequent views acquired very close in time with a very small motion among them.

Recently, a new “distance formulation” has been proposed [112] where the model surface is implicitly represented as the zero isosurface of a fitted Radial Basis Func-

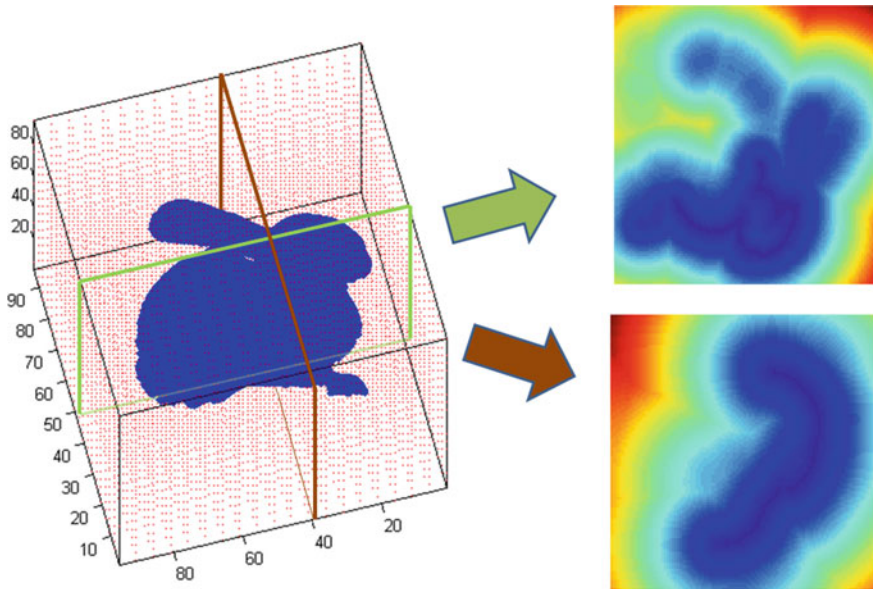


Fig. 8.5 Using the distance transform. The model-view is enclosed in a volumetric grid (left). For each point of the grid, the closest model point is computed. Two planes are highlighted on the XY - and YZ -axes, respectively, and the distance transform values of each grid point are visualized for both planes (right)

tion (RBF), $s(\mathbf{x}) = 0$, for any 3D point \mathbf{x} , where the function s represents *distance to surface*. For any point on the data scan (or on a pre-computed 3D grid), the distance and direction (gradient) to the zero isosurface can be computed directly from the RBF. The advantage of this RBF distance formulation is that it interpolates over holes that may exist in the model scan. Particularly for lower resolution scans, the interpolation is more accurate than the piecewise linear point-to-plane method. Both RBF model fitting and RBF model evaluation are $O(n \log n)$.

8.2.3.3 Techniques for Improving Accuracy

The accuracy of the alignment is the most critical aspect of the registration since even a small misalignment between two views can affect the whole 3D model reconstruction procedure. The simplest strategy that can be used is outlier rejection. Other methods improve the accuracy by using additional information such as color and texture or local geometric properties. Finally, an effective class of methods devoted to the improvement of accuracy are probabilistic methods.

Outlier Rejection

Closest point computation may yield spurious correspondences due to errors or to the presence of non-overlapping parts between the views. Typically, outlier rejection

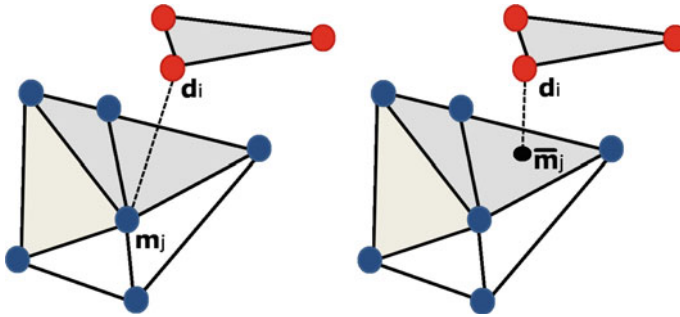


Fig. 8.6 Distance formulation. Point-to-point distance: the 3D vertex d_i is associated to the 3D point m_j which is a vertex of the source 3D mesh (left). Point-to-plane distance: the 3D vertex d_i is associated to the 3D point m_j which lies inside the plane defined by a triangle of the source 3D mesh (right)

techniques threshold the residuals. The threshold can be fixed manually, or as a percentage of worst pairs (e.g., 10% [117, 126]). Other techniques perform statistics on the residual vector and set the threshold as 2.5σ or apply the so-called *X84* rule [29, 66]. An evaluation of the use of the *X84* rule for automatic outlier rejection is presented in Sect. 8.5. More recently, statistical analysis has been introduced into the general registration problem (Eq. 8.1) by proposing a new error function named *Fractional Root Mean Squared Distance* [113]. In [17], an implicit approach to reject the outliers is introduced exploiting a sparse formulation of the closest point computation with L_1 -norm distance.

Additional Information

The basic ICP algorithm computes the correspondences by taking into account only the *proximity* of points. However, corresponding points should be similar with respect to other aspects. Several studies have attempted to exploit additional information available from the acquisition process or from the analysis of the surface properties. In practice, the distance formulation is modified to integrate such additional information like local surface properties [59], intensity derived from the sensor [59, 154], or color [118]. In [74], the authors proposed to use color and texture information. In [135], the so-called *ICP using Invariant Feature* (ICPIF) was introduced where several geometric features are employed, namely, *curvatures*, *moments invariants*, and *spherical harmonics invariants*. In [25], additional information was integrated into the point descriptors using the spin image with color. More recently, with the availability of low-cost RGBD sensors [165], several methods have been proposed that exploit the matching computation on both the 2D and 3D domains (i.e., a 2D image is used as additional information) [67]. For instance, [162] uses camera pose optimization with 2D features to improve the accuracy of registration.

Probabilistic Method

In order to improve the robustness of the registration, several probabilistic versions of the standard ICP have been proposed [61, 119, 120]. In [119, 120], the idea

of multiple weighted matches justified by a probabilistic version of the matching problem is introduced. A new matching model is proposed based on Gaussian weight (SoftAssign [120]) and mutual information [119], leading to a smaller number of local minima and thus presenting the most convincing improvements. In [61], the authors introduced a probabilistic approach based on the Expectation Maximization (EM) paradigm, namely, EM-ICP. Hidden variables are used to model the point matching. Specifically, in the case of Gaussian noise, the proposed method corresponds to ICP with multiple matches weighted by normalized Gaussian weights. In practice, the variance of the Gaussian is interpreted as a scale parameter. At high scales, EM-ICP gets many matches while it behaves like standard ICP at lower scales. In [47], a hierarchical approach was introduced representing the partial views at multiple scales. In this fashion, the most appropriate level of geometric details is adaptively found to improve the point matching.

8.3 Advanced Techniques

Although registration is one of the most studied problems in computer vision, several cases are still open and new issues have emerged in recent years. In this section, we focus on some scenarios where registration becomes more challenging: registration of *more than two views*, registration in *cluttered scenes*, and registration of *deformable objects*. We also describe some emerging techniques based on machine learning to solve the registration problem. Figure 8.7 illustrates the proposed taxonomy for advanced registration techniques.

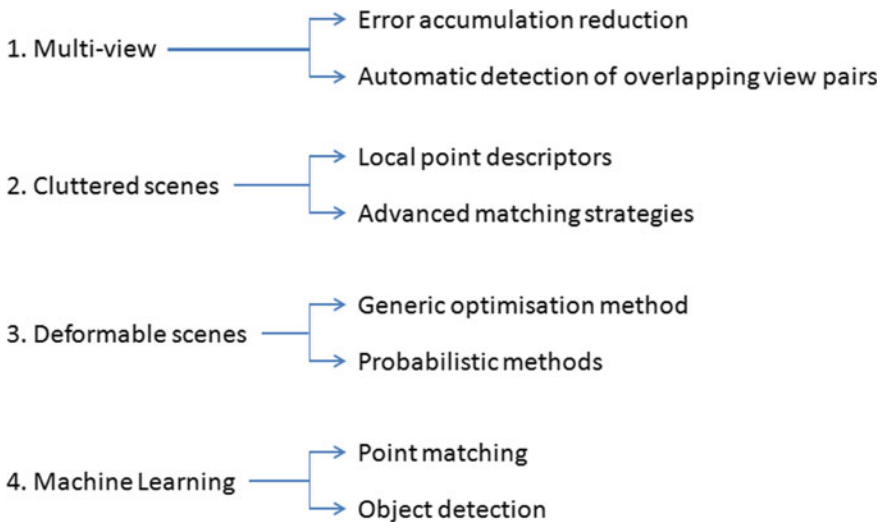


Fig. 8.7 A taxonomy of advanced registration techniques

8.3.1 Registration of More Than Two Views

Once registration has been performed pairwise, all the views need to be transformed into a global reference system by applying a *multiple-view* registration technique. Note that in this context the assumption that the data-view is a subset of the model-view is no longer true. There are two main issues: (i) error accumulation and (ii) the automation of the process.

Reducing Error Accumulation

When the ordering of the sequence of views N_1, \dots, N_p is available, the registration can be performed pairwise between consecutive views (i.e., between views N_i and N_{i+1}). In general, even if all the pairs are apparently well registered, some misalignment typically appears when the full model is reconstructed due to the accumulation and propagation of the registration error. The general idea of multiple-view registration techniques is to solve *simultaneously* for the global registration by exploiting the interdependencies between all views at the same time. This introduces additional constraints which reduce the global error. A comparative study of similar multiple-view registration schemes was performed [43]. In [117], a method is presented that first aligns the scans pairwise with each other and then uses the pairwise alignments as constraints in a multiview step. The aim is to evenly distribute the pairwise registration error, but the method itself is still based on pairwise alignments. In [29], a method that distributes registration errors evenly across all views was proposed. It operates in the space of estimated pairwise registration matrices; however, ordering of the views is required. More recently, [144] proposed a new approach based on the well-known generalized Procrustes analysis, seamlessly embedding the mathematical theory in an ICP framework. A variant of the method, where the correspondences are non-uniformly weighted using a curvature-based similarity measure, was also presented. In [161], a method that handles loop closures was proposed to perform a globally consistent reconstruction. Locally fused models are introduced for overlapping parts of the scene and used to initialize a global graph-based optimization that distributes residual error. The key idea consists in the detection of *points of interest* characterized by the areas with the highest density of information. This approach is particularly effective for large-scale scenarios where the reconstruction is obtained with a SLAM-like framework [51]. In [53], error accumulation is avoided by extending the LM-ICP algorithm [54] to work on multiple views. The idea consists of defining an effective optimization function that considers all the views simultaneously in the registration error, whose solution is obtained using standard numerical methods.

Automating Registration

Especially when the full model is composed of a large number of scans, the view order might not be available and therefore should be manually specified. Many methods were proposed to improve the automation of multiple-view registration. In [72], a global optimization process searches a graph constructed from the pairwise view matches for a connected sub-graph containing only correct matches, using a global

consistency measure to eliminate incorrect but locally consistent matches. Other approaches use both *global* and *local* pre-alignment techniques to select the overlapping views by computing a coarse alignment between all the pairs. In [90], the pre-alignment is performed by extracting global features from each view, namely, extended Gaussian images. Conversely, in [78], the pre-alignment is computed by comparing the signatures of feature points. Then, the best view sequence is estimated by solving a standard Travelling Salesman Problem (TSP). In [37], a robust strategy was introduced to detect wrong alignment between view pairs. A global optimization method is introduced based on the line processes algorithm. In [65], the authors introduced a *shape-growing* method where a seed shape is sequentially updated by registering it with the input partial views. In [53], a view matching method was proposed that combines salient point descriptors with a RANSAC-based robust correspondence computation. In this fashion, both the ordering of views and the pairwise pre-alignment are obtained leading to a fully automatic registration pipeline. Results regarding this aspect for example 3D scans are presented in Sect. 8.4.

8.3.2 Registration in Cluttered Scenes

Thanks to the recent availability of large-scale scanners, it is possible to acquire scenes composed of several objects. In this context, registration is necessary to localize each object present in the scene and estimate its pose. We call this scenario a *cluttered* case where the overlap between the registering views is very small since the object of interest to be localized may be made of a small subset of the entire view. This makes the registration problem more challenging. Figure 8.8 shows two examples of highly cluttered scenes: an entire square² and a scene composed of several mechanical objects.

Roughly speaking, two main strategies were proposed to address this problem: (i) the use of point signatures to improve point-to-point matching and (ii) the design of more effective matching methods.

Point Signatures

This approach is similar to local approaches for pre-alignment. However, in cluttered scenarios, the problem becomes more challenging since, differently from the standard pre-alignment case, the neighborhood of one point of an object can cover part of other objects. Therefore, the descriptor may become useless, and the size of local neighborhood becomes crucial to get the best tradeoff between reliability of descriptor and robustness to clutter. A large number of methods for keypoint detection were proposed to reduce the considered points only to very few salient areas [145]. Then, using a local 3D Reference Frame (RF) is important to encode pose invariant 3D descriptors. For instance, in [146] the so-called *SHOT* descriptor was proposed to form a rotation invariant and robust to noise RF from which a descriptor is obtained combining geometric information with color. In [95], a descriptor

²Piazza Brà, Verona, Italy. Image courtesy of Gexcel: <http://www.gexcel.it>.



Fig. 8.8 Example of large scan acquisition (left) and scene with multiple mechanical objects (right)

that uses two reference points to define a local coordinate system is proposed. In particular, a three-dimensional tensor is built by sampling the space and storing the amount of surface intersecting each sample. In [8], a method that exploits surface scale properties is introduced. The geometric scale variability is encoded in the form of the intrinsic geometric scale of each computed feature by leading to a highly discriminative hierarchical descriptor.

Matching Methods

Since the number of corresponding points is very few within cluttered scenes, standard methods for outlier rejection are not useful but more complex matching algorithm can be exploited. In [95], descriptors are stored using a hash table that can be efficiently looked up at the matching phase by geometric hashing algorithm. In [8], matching is performed in hierarchical fashion by using the hierarchy induced from the definition of point descriptor. In [46], a method is proposed that creates a global model description using an oriented point pair feature and matches it by using a fast voting scheme. A fast voting scheme, similar to the generalized Hough transform, is used to optimize the model pose in a locally reduced search space. This space is parametrized in terms of points on the model and rotation around the surface normals.

In [155], a new voting scheme called *Intrinsic* Hough transform was introduced to exploit the sparsity of the voting space by sampling only at the areas where the matching probability is non-zero. In [2], a method is proposed to extract all coplanar 4-point sets from a 3D point set that are approximately congruent, under rigid transformation, to a given set of coplanar 4-points. This approach is further expanded in the so-called *Super4PCS* method [93] where an effective data structure is exploited to improve the core *instance problem*, i.e., finding all point pairs that are within a distance range $(r - \epsilon, r + \epsilon)$. In this fashion, a very fast registration can be obtained from arbitrary pose, with very few overlap. In [163], a method called *fast global* registration was introduced. A well-defined energy formulation is designed to encode the registration constraints, and a line process technique is exploited to

efficiently solve for the optimal solution. This method is suitable for large scenes with clutter. Moreover, it can easily work with multiple views.

8.3.3 Deformable Registration

While rigidity in the aligning transformation is a largely applicable constraint, it is too restrictive in some cases. Imagine indeed that the object that has to be registered is not rigid but deformable. For instance, a typical deformable object is the human body and its parts such as the face or the hands. For the full body, it is very important to align the articulated parts (i.e., arms, legs, and so on), while for the face the deformations are caused mainly by the facial expressions. Another class of object is composed of planar shapes such as a piece of paper or a blanket that deform over time. Also, in the medical domain, there are mainly non-rigid scenarios caused by the deformation of the internal parts of the human body. Deformable registration has two main issues: the computation of stable correspondences and the use of an appropriate deformation model. Note that the need for registration of articulated or deformable objects has recently increased due to the availability of real-time range scanners [33, 34, 85, 97]. Roughly speaking, we can emphasize two classes of deformable registration methods: (i) methods based on general optimization techniques, and (ii) probabilistic methods.

Methods Based on General Optimization Techniques

The general formulation of deformable registration is more involved than the rigid case and it is more difficult to solve in closed form. Advanced optimization techniques are used instead. The advantage of using general optimization techniques consists of jointly computing the estimation of correspondences and the deformable parameters [33, 34, 38, 85, 138]. Moreover, other unknowns can be used to model further information like the overlapping area, the reliability of correspondences, the smoothness constraint, and so on [85]. Examples of transformation models which have been introduced for surface deformations are (i) affine transforms applied to nodes uniformly sampled from the range images [85], (ii) rigid transforms on patches automatically extracted from the surface [33], (iii) as rigid as possible constraint [138], (iv) Thin-Plate Splines (TPS) [38, 124], or (v) Linear Blend Skinning (LBS) model [34]. The error function can be optimized by the Levenberg–Marquardt Algorithm [85], GraphCuts [33], or Expectation–Maximization (EM) [34, 38, 100]. In [71], deformable registration is solved by alternating between correspondence and deformation optimization. Assuming approximately isometric deformations, robust correspondences are generated using a pruning mechanism based on geodesic consistency. Deformable alignment to account for errors in the point clouds obtained by scanning a rigid object is proposed in [22, 23]. Also, in this case, the authors use TPS to represent the deformable warp between a pair of views that they estimate through hierarchical ICP [124]. Note that using real-time RGBD sensors is possible to implement real-time modeling and reconstruction systems also for non-rigid objects

[73, 102]. For instance in [102], the so-called *Dynamic Fusion* method has been introduced for dense dynamic scene reconstruction. The idea is to estimate a dense volumetric motion field for each frame in order to provide increasingly denoised measurements and a complete representation of the observed scene as more measurements are acquired and integrated. This approach has been further extended in [73] where a sparse RGB feature matching strategy was introduced to improve the robustness of tracking.

A new class of methods are properly defined for human shapes and are based on the fitting of a template model (e.g., a morphable model) to the acquired shape [88]. In this scenario, the challenge consists of estimating both the shape and pose parameters within the same optimization procedure. For instance, in [14], the Skinned Multi-Person Linear (SMPL) model is registered to a monocular RGBD sequence for full-body reconstruction. This approach was further expanded for full-body dynamic shape and motion capture [15] where a new mesh registration method was proposed that uses both 3D geometry and texture information to register all scans in a sequence to a common reference topology. Another class of important method for non-rigid shapes is based on spectral shape analysis [84]. In particular, a new approach consists of moving the computation of matching from the physical to the spectral space exploiting the so-called *functional map* framework [106]. From the basic framework, several variants have been proposed to work on partial shapes [87, 122], to reduce the number of estimated parameters [103], and to exploit locality on the spectral domain [94]. It is worth noting that these methods provide the matching between points or shape parts but they cannot deform the pair of shapes to allow a full alignment between them. For this aim, a recent method called functional automatic registration for 3D human bodies (FARM) was proposed in [91] where the spectral approach based on the functional map is combined with the SMPL template model to obtain the registration of the full body in very challenging scenarios such as partiality, noise, and topological variation.

Probabilistic Methods

Using probabilistic methods, the uncertainty on the correct surface transformation can be addressed by adopting maximum likelihood estimation [4, 45, 70, 75, 100, 151]. Probabilistic approaches are based on modeling each of the point sets by a kernel density function [148]. The dissimilarity among such densities is computed by introducing appropriate distance functions. Registration is carried out without explicitly establishing correspondences. Indeed, the algorithm registers two meshes by optimizing a joint probabilistic model over all point-to-point correspondences between them [4]. In [75], the authors propose a correlation-based approach [148] to point set registration by representing the point sets as Gaussian mixture models. A closed-form solution for the L_2 norm distance between two Gaussian mixtures makes fast computation possible. In [151], registration is carried out simultaneously for several 3D range datasets. The method proposes an information-theoretic approach based on the Jensen–Shannon divergence measure. In [100], deformable registration is treated as a maximum likelihood estimation problem by introducing the coherent point drift paradigm. Smoothness constraints are introduced based on the assumption

that points close to one another tend to move coherently over the velocity field. The proposed energy function is minimized with the EM algorithm. Similar approach has been proposed in [45] to track the full-hand motion. A stereo setup is employed to estimate the 3D surface. To improve the estimation of the hand pose, 2D motion (i.e., optical flow) is combined with 3D information. A well-defined hand model is employed to deal with articulated structures and deformations. Also, in this case, the standard ICP algorithm has been extended to its probabilistic version according to the EM-ICP approach. This approach has been further extended in [70] where the so-called expectation *conditional* maximization paradigm is introduced. A formal demonstration is proposed to show that it is convenient to replace the standard M-step by three conditional maximization steps, or CM steps, while preserving the convergence properties of EM. Experiments are reported for both the hand and body tracking. In [28], a statistical method has been proposed to model the local geometry properties variation as a local stochastic process encoded in a Hidden Markov Model (HMM). The idea is to learn local geometric configurations as hidden states and encoding the surface properties in terms of transitions among such states. In [166], the so-called *Stitched Puppet* model was proposed to encode the body parts of the human shapes in a generative model. The human body is represented by a graphical model whose nodes are associated to body parts that can independently translate and rotate in 3D.

8.3.4 Machine Learning Techniques

Recently, advanced machine learning techniques have been exploited to improve registration algorithms [3, 60, 98, 111, 139]. The general idea is to use data-driven approaches that learn the relevant registration criteria from examples. The most promising methods have been proposed for (i) improving the matching phase and (ii) detecting an object which is a general instance of one or more classes. Most of the recently proposed methods are based on deep learning architectures [82].

Improving the Matching

In these approaches, the emphasis is on the effectiveness of the correspondence computation. In [139], a new formulation for deformable registration (3D faces) is proposed. The distance function from corresponding points is defined as a weighted sum of contributions coming from different surface attributes (i.e., proximity, color/texture, normals). Instead of manually or heuristically choosing the weights, a machine learning technique is proposed to estimate them. A support vector machine framework is employed in a supervised manner, based on a dataset of pairs of correct and incorrect correspondences. In [3], the authors propose a novel unsupervised technique that allows one to obtain a fine surface registration in a single step, without the need of an initial motion estimation. The main idea of their approach is to cast the selection of correspondences between points on the surfaces in a *game theoretic* framework. In this fashion, a natural selection process allows one to select points

that satisfy a mutual rigidity constraint to thrive, eliminating all the other correspondences.

With the explosion of the deep learning technology [82], several methods have been proposed to extend this very effective approach to 3D registration. In [158], a neural network is trained on 3D shapes working on local volumetric patches. The need of a large number of training data is satisfied using the available dataset of already registered scans. This approach was further improved in [44], encoding simple geometric relationships such as normals and point pair features to better represent the local context of a given point. Furthermore, to explore the relationship across different views, a new descriptor was proposed in [160] where training is carried out by collecting information from multiple views. In [50] rather than using labeled data, the authors proposed an unsupervised method based on deep auto-encoders [68]. Deep learning methods are also effective for non-rigid objects like the human body [16, 86]. In [16], a geometric convolutional neural network was designed to effectively learn class-specific shape descriptors. In [86], a new neural network was proposed to directly estimating the shape correspondences within the functional map framework. Finally, interesting deep learning approaches have been successfully employed in the medical domain where different sources of information need to be integrated using a multimodal registration procedure [137].

Object Detection

A new class of methods is emerging from employing machine learning techniques for detecting specific classes of objects on large scenes [60, 98, 111]. In this context, the registration is important to be able to devise a *detection-by-localization* approach [147] where the pose of the object is also estimated. Several works have been done for the 2D domain, but its extension to 3D scenes is not trivial. In [111], the authors proposed to detect cars in cluttered scenes composed of millions of scanned points. The method is based on integrating *spin images* with *extended Gaussian images* in order to combine effectively local and global descriptors. Furthermore, the method is able to detect object classes and not only specific instances. In [98], the Associative Markov Network (AMN) has been extended to integrate the context of local features by exploiting directional information through a new non-isotropic model. In [60], different objects are simultaneously detected by hierarchical segmentation of point clouds. Indeed, clusters of points are classified using standard learning by example classifiers.

Deep learning methods have shown their benefit also for the object detection task. The so-called *3DMatch* method [158] enables 3D model alignment from cluttered RGBD scenes using a deep neural network to learn 3D matching. In [81], the object is localized using a regression procedure which, from the depth image, provides the 6D-pose parameters. Note that these deep-learning-based approaches naturally extend to multiple and different objects, leading to a complete 3D semantic segmentation. For instance, in [121] a new neural network architecture was proposed for the extraction of semantic components of a point cloud.

8.4 Registration at Work

In this section, we show some practical examples of registration algorithms at work on two views and then on more than two views.

8.4.1 Two-View Registration

As mentioned in Sect. 8.2, given a pair of partial views of the same object, the pairwise registration task consists in estimating the rigid transformation that brings the moving *data-view* to the reference *model-view*. In this example, we take a pair of scans from an RGBD sequence of a real scene that is acquired by a real-time sensor (i.e., kinect [101]). The scene contains a table and some objects of the real life. The data is online available from the *rgbd-scene-v2* dataset³ and represents a very useful benchmark for the evaluation of real-time modeling techniques like *kinect fusion* [101].

Figure 8.9 shows the pair of evaluated scans. The RGB image (Fig. 8.9, left) shows a can, a box, a bowl, and a mug. The depth map (Fig. 8.9, center) highlights the relative positions among the involved objects (i.e., the bowl is the closest to the observer, the box is the farthest, and the other objects are in between). The colored point cloud (Fig. 8.9, right) gives the full 3D representation of the scene. As happens usually for a real scene acquired with real-time RGBD sensors, the scans are very noisy with the presence of many holes and outliers.

Figure 8.10 shows the 2D registration performance. Figure 8.10 shows the partial views before the registration. The second view is acquired after around 1 s from the first one. This means that the sensor motion is sufficiently large to introduce new details of the overall scene, and the two views are already heavily misaligned. For this experiment, we employed the standard ICP algorithm for pairwise registration after a manual pre-alignment. We use the ICP implementation from meshlab⁴ and the enclosed aligning toolkit for the pre-registration. Finally, Fig. 8.10 shows the registered views. The two views are correctly aligned, and new details of the acquired scene can be correctly captured especially on the can and the bowl.

8.4.2 Multiple-View Registration

In this section, we evaluate a fully automatic 3D registration pipeline of multiple views. Given a sequence of unordered partial views of the same object, the registration task consists in finding the rigid transformation that brings each view to the global reference system. In this fashion, the views become aligned and a com-

³<http://rgbd-dataset.cs.washington.edu/dataset/rgbd-scenes-v2/>, scene 13, frames 200 and 230.

⁴<http://www.meshlab.net/>.

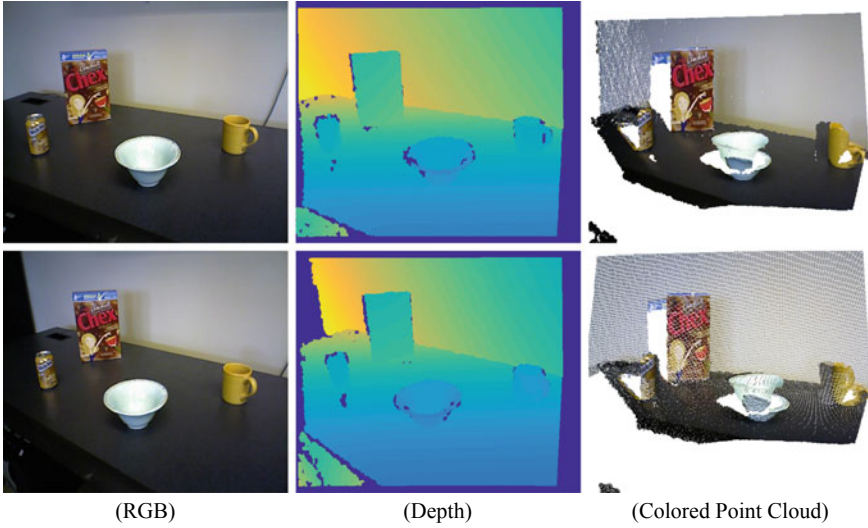


Fig. 8.9 Pair of RGBD scans: RGB image (left), depth map (center), and colored point cloud (right)



Fig. 8.10 2D Registration: starting views (left) and registered views (right)

plete model can be reconstructed. We adopt a local feature-based approach which is composed of the following main steps⁵:

1. **Keypoint detection.** Keypoint detection aims at detecting few and salient feature points from the shape. We employ the salient point detection method proposed in [27]. Inspired by the research on saliency measure on 2D images, the source mesh is decomposed into multiscale representations, and a saliency measure is defined by combining the results gathered at each scale. Finally, maximal points on the salient map are detected as feature points.

⁵Code and data for this automatic 3D registration pipeline is implemented on the Automatic 3D Registration toolkit available at <http://profs.sci.univr.it/~castella/art.html>.

2. **Keypoint description.** Keypoint description aims at attaching a descriptor to each keypoint that must be (i) distinctive of the point, (ii) invariant to rigid transformations, and (iii) resilient to as much nuisances as possible (noise, clutter, partial views, sampling rate, and so on). We use spin images [76], a well-known surface representation that has been successfully used in shape matching and object recognition.
3. **View matching.** Since the ordering of the views is unknown, a view matching step is required to estimate the set of view pairs to be registered. To this aim, the overlap between each view pair is computed using a voting scheme among the keypoint descriptors. The output of this stage is encoded in an adjacency matrix between the views.
4. **Pairwise registration.** The pairwise registration is obtained in two phases: (i) robust pre-alignment for which only feature points are involved, and (ii) ICP registration refinement where a more accurate alignment is estimated with ICP on the pre-aligned views using only the overlapping parts.
5. **Global registration.** A global alignment is produced by combining the pairwise rigid transformations found in the previous step. The idea (as in [92, 132]) is to estimate a global alignment of the views with the least accumulation error among the solutions based on chaining pairwise registrations.

A more detailed description of the steps involved in the proposed pipeline is available in [53]. Figure 8.11 shows a subset of partial views of Bunny (i.e., 6 over 24 views). The overlap between views is reliable for only a few view pairs (e.g., views 1 and 5). Conversely, for most of the view pairs, the overlap is not sufficient to guarantee a correct alignment. According to our pipeline, we detect the keypoints and their signatures for each view. Figure 8.12 (top) and (middle) shows some feature points on the Bunny ear from two overlapping views. It is worth to note that the extracted keypoints are coherent on the two observed views. Figure 8.12 (on the right side) shows the spin images of selected keypoints (red dots). As expected, corresponding points generate very similar signatures (Fig. 8.12 (top) and (middle) on the right side). When instead we consider a pair of non-corresponding points, their signatures appear very different (Fig. 8.12 (bottom) where a point around the eye is observed).

To compute the view matching, we follow the approach of [24] for 2D image mosaicing. In this phase, we consider only a constant number of descriptors in each view (we used 100, where a typical view contains thousands of keypoints). Then, each keypoint descriptor is matched to its l nearest neighbors in feature space (we use $l = 6$). This can be done efficiently by using a k-d tree to find approximate nearest neighbors. A 2D histogram is then built that records in each bin the number of matches between the corresponding views: we call it the keypoint co-occurrence matrix H . Finally, every view is matched to the $m (= 8)$ views that have the greatest values in H . Figure 8.13 (left) shows the keypoint co-occurrence matrix for our 24 views.

Once the set of overlapping views is available, we proceed with a robust pairwise registration procedure. At the first stage, a point-to-point matching is computed

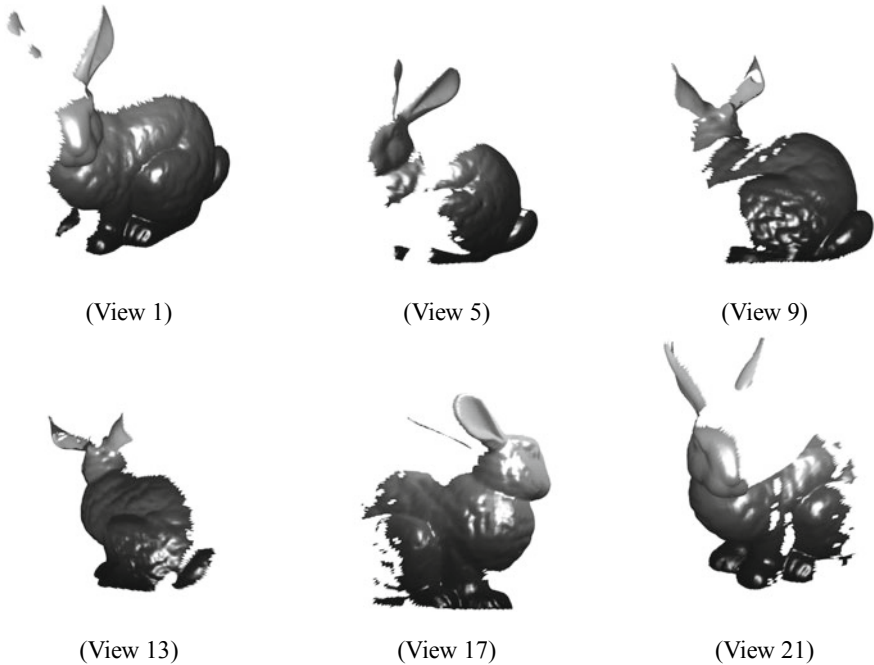


Fig. 8.11 Bunny. The full model is observed from partial views gradually sampled around the object (front to back views 1, 5, 9, and back to front views 13, 17, 21). Only 6 over 24 available views are shown (i.e., one every four views)

between keypoints using a nearest neighbor strategy among feature descriptor. Then, a geometric constraint is introduced using RANSAC on the absolute orientation. This leads to a robust pre-aligned between the views that are further refined using ICP. Figure 8.14 shows two overlapping views before (left) and after (right) registration. The overlapping points are colored in green. Note that for partially overlapping views, the inliers correspond to the area of overlap; hence, we can assign a weight $W(i, j)$ in the range $[0, 1]$ to the pair (view i , view j), corresponding to the fraction of the overlapping points over the total number of points. The $n \times n$ matrix W is called the weighted adjacency matrix. For our experiment, the adjacency matrix is shown in Fig. 8.13 (right).

Finally, a weighted graph is constructed, whose vertices are the views and whose (weighted) adjacency matrix is W . Given a reference view chosen arbitrarily, which sets the global reference frame, for each view i , the transformation that aligns it with the reference view r is computed by chaining transformations along the shortest weighted path from i to r . This is equivalent to computing the (weighted) Minimum Spanning Tree (MST) with the root in r . In our experiment, the reference view is $r = 1$. Figure 8.15 shows the viewgraph (top) and the extracted path (bottom). In this fashion, the correct order for chaining the transformations is recovered, and all the

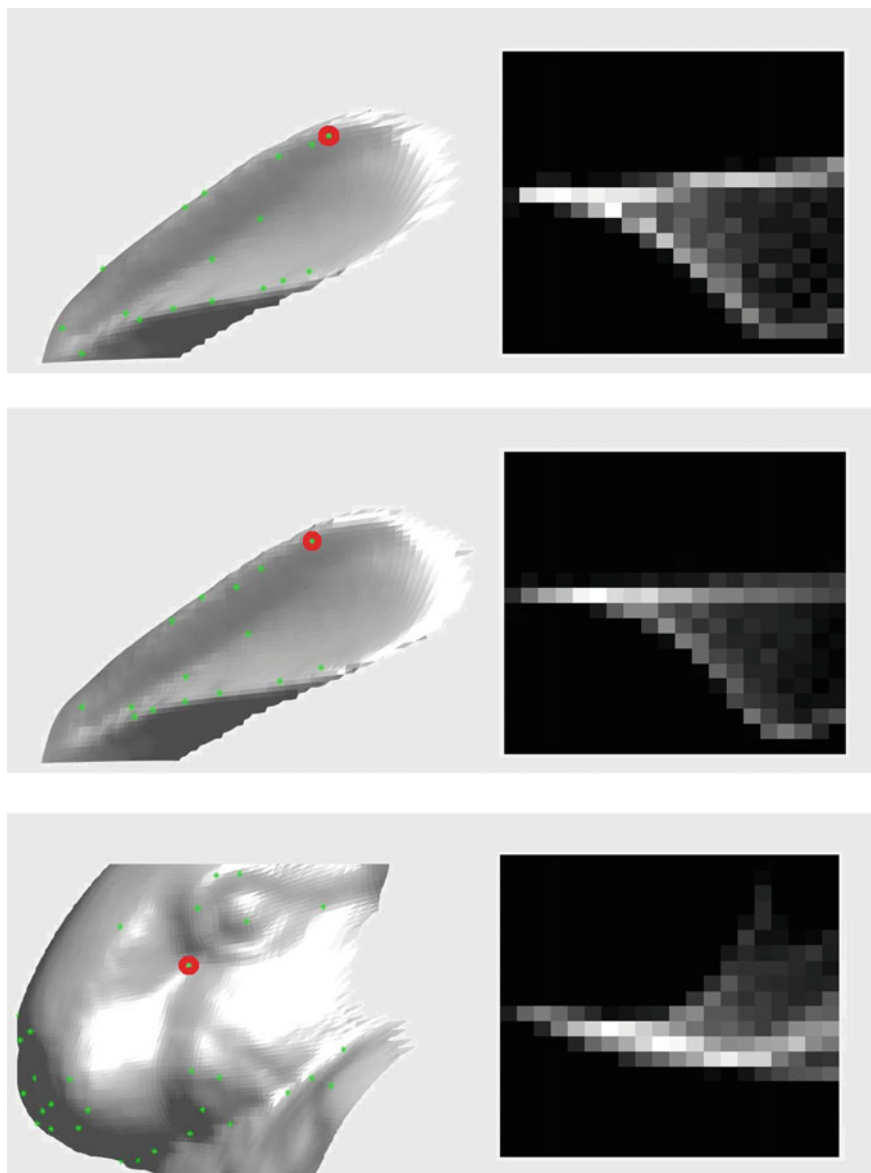


Fig. 8.12 Keypoint and signature. Keypoints are extracted from a small subpart of Bunny (i.e., the ear from two different views—top and middle—and the eye—bottom). For red points, the spin image signature is shown on the right

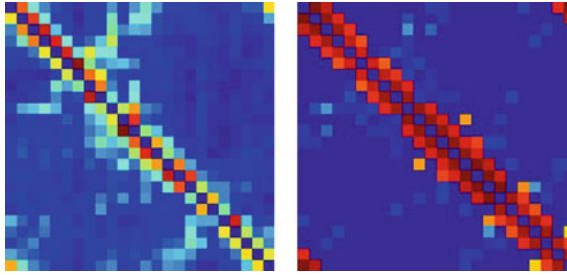


Fig. 8.13 The keypoint co-occurrence matrix H (left) and the adjacency matrix W (right)

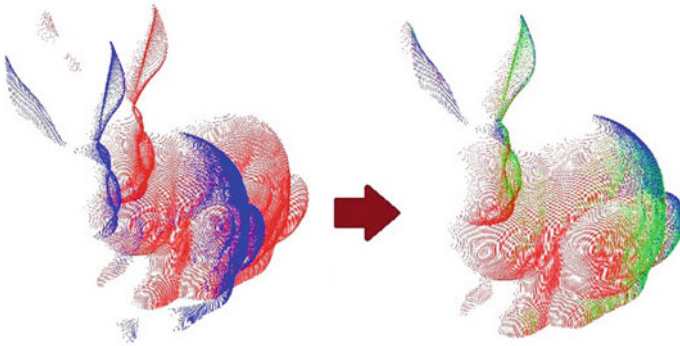


Fig. 8.14 Pairwise registration. Two partial views are shown before (left) and after (right) registration. The overlap is colored in green

views can be represented on the global reference system. Figure 8.16 shows the views before (left) and after (right) the multiview registration. All the views are correctly aligned, and a full model of the observed Bunny can be reconstructed.

8.5 Case Study 1: Pairwise Alignment with Outlier Rejection

In this section, we describe a simple but effective strategy to make the ICP algorithm resistant to wrong correspondences. Especially when views are only partially overlapped, many points of the data-view do not have a correspondence in the model-view. We call those points *single points*. However, the basic ICP enforces single points to be associated to closest points in the model-view, therefore generating outliers. A robust outlier rejection procedure is introduced based on the so-called *X84* rule [29, 66]. The idea is to perform a robust statistical analysis of the residual errors e_i after closest point computation. The underlying hypothesis was pointed out in [159] and consists of considering the residuals of two fully overlapping sets as an approxima-

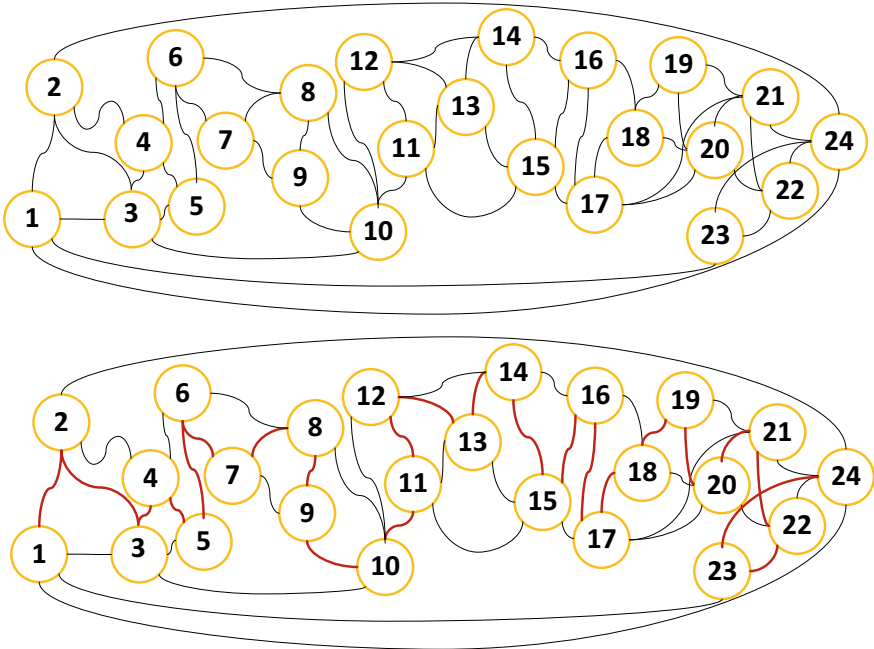


Fig. 8.15 The viewgraph is obtained from the adjacency matrix W (top). From the graph, the sequence of chaining transformations is computed using the minimum spanning tree algorithm (bottom). The root is node 1

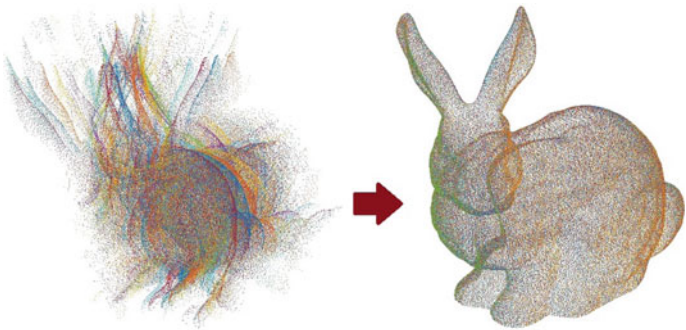


Fig. 8.16 Multiview registration. All 24 views before the registration (left) and after registration to the global reference system (right)

tion of a Gaussian distribution. Non-overlapping points can be detected by estimating a Gaussian distribution from residual errors and by defining a threshold on the tails of the estimated Gaussian.

The X84 rule is a tool to estimate robustly and automatically this threshold. Given the residual errors $\mathbb{E} = \{e_i\}$, $i = 1 \dots N_d$, the *Median Absolute Deviation* (MAD) is defined as

$$MAD = med(|e_i - location|), \quad (8.10)$$

where *med* is the *median* operator and *location* is the median of residual errors (i.e., $med(\mathbb{E})$). The X84 rule prescribes to reject values that violate the following relation:

$$|e_i - location| < k \cdot MAD. \quad (8.11)$$

Under the hypothesis of Gaussian distribution, a value of $k = 5.2$ is adequate in practice, as the resulting threshold contains more than 99.9% of the distribution.

Now we are ready to define the new procedure for robust outlier rejection:

1. For all data point $\mathbf{d}_i \in D$, compute the error e_i according to Eq. 8.4 (i.e., by estimating the closest point and by generating the pair of corresponding points $\mathbf{c}_i = (\mathbf{d}_i, \mathbf{m}_j)$).
2. Estimate *location* by computing the median of residuals $med(\mathbb{E})$.
3. Compute *MAD* according to Eq. 8.10.
4. For each residual error e_i ($i = 1, \dots, N_d$):
 - a. If e_i satisfies Eq. 8.11 then keep \mathbf{c}_i in the list of correspondences,
 - b. If not, reject the correspondence.
5. A new list of corresponding points $\hat{\mathbf{c}}_i$ is obtained from which outliers have been filtered out.

In practice, this procedure replaces step 1 in the ICP algorithm described in Sect. 8.2.2. The X84 rejection rule has a breakdown point of 50%: any majority of the data can overrule any minority. The computational cost of X84 is dominated by the cost of the median, which is $O(n)$, where n is the size of the data point set. The most costly procedure inside ICP is the establishment of point correspondences, which costs $O(n \log n)$. Therefore, X84 does not increase the asymptotic complexity of ICP.

In Fig. 8.17, an example of registration between two views with a strong occluded part is shown. The non-overlapping area is wide: the ears and the whole face of *Bunny* are only visible in the data-view, while the bottom part of the body is observed in the model-view only. The number of data point is $N_d = 10000$, the number of model point $N_m = 29150$, and the number of points of the overlap is $\#(\mathbb{D} \cap \mathbb{M}) = 4000$. In this experiment, the two views are synthetically sampled from the whole 3D model. A view mask of 600×500 points is used in order to obtain highly dense views. Moreover, in this fashion we know the ground truth transformation, and no noise affects the views. Figure 8.18 shows the distribution of residual errors after X84-ICP registration. Note that most of the residuals are concentrated around zero.

Table 8.1 X-84 performance evaluations. Rotation and translation errors are reported

| Method | Rot-error (rad.) | Transl-error (mm) | # Overlap. points | # Iterations | Time (s) |
|--------------|------------------|-------------------|-------------------|--------------|----------|
| Besl [11] | 0.22345 | 1.2636 | 10000 | 20 | 370 |
| Picky [164] | 0.10918 | 0.9985 | 9534 | 28 | 76 |
| X84-ICP | 0.06351 | 0.4177 | 4582 | 21 | 383 |
| Ground truth | – | – | 4000 | – | – |

It is confirmed that the behavior of the early part of the distribution is similar to a Gaussian [159]. The X84 rule is employed, and the threshold is automatically estimated on the tail of the Gaussian. The second peak of the distribution corresponds to residuals generated by the non-overlapping points.⁶ In Fig. 8.18 (right), points of the data-view are colored differently between inliers and outliers. Note that non-overlapping parts are correctly registered.

Table 8.1 summarizes the performance of X84-ICP in comparison with Besl, i.e., the standard ICP [11], and Picky [164] which implements a combination of ICP variations described in Sect. 8.2.3. A hierarchical sampling strategy is introduced to improve the speed, and a thresholding approach on the residual distribution is employed. More specifically, a threshold is defined as $TH = \mu + 2.5\sigma$, where $\mu = \text{mean}(\{e_i\})$ and $\sigma = \text{std}(\{e_i\})$. The ground truth transformation is shown as well. Note that the basic ICP is strongly affected by outliers and is not able to correctly align the two views. The Picky ICP improves the accuracy, but it is not able to correctly

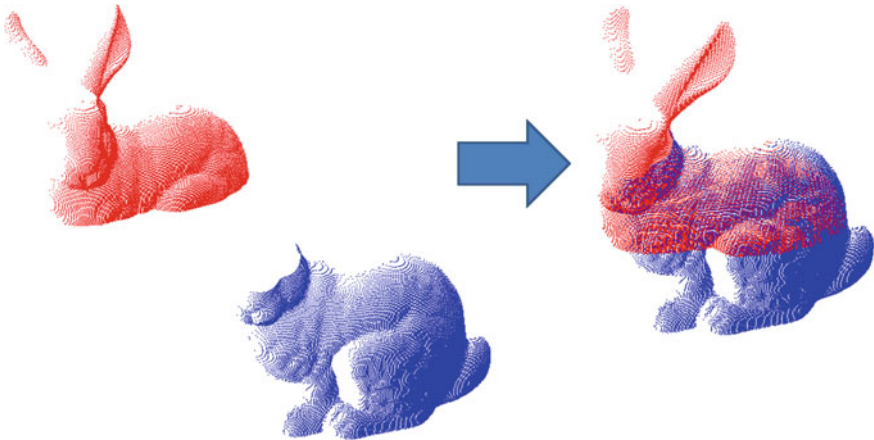


Fig. 8.17 Registration with robust outliers rejection. Two views at starting pose (left) and after registration (right). Note that the overlap area is quite restricted

⁶In order to visualize the peak, the second part of the histogram has been quantized with wider intervals.

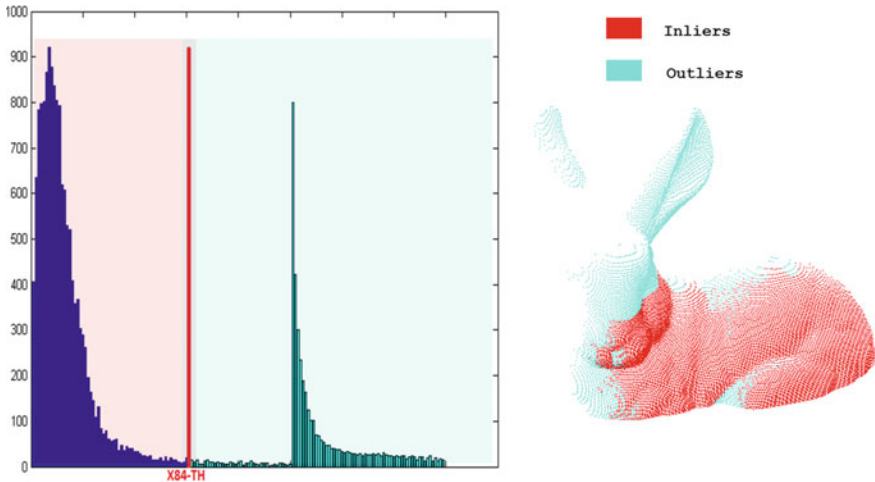


Fig. 8.18 Automatic residuals thresholding. From the distribution of residuals, the threshold is estimated according to the X84 rule. Points under the threshold are inliers (red), while outliers are over the threshold (blue). Outliers are points in non-overlapping areas

estimate the overlapping parts and it does not reach convergence. Conversely, by employing the X84 rule, wrong correspondences are well detected and a correct registration is obtained. In order to see the improvement in using robust methods, we evaluate the X84-ICP on a more challenging scenario composed of a pair of very noisy images with low resolution. The scene represents a tubular structure in an underwater environment that is acquired using acoustic devices [29, 30]. The starting views are depicted in Fig. 8.19 (left). In the red view, we clearly see a large amount of outliers in the left part of the scene. Figure 8.19 (center) shows the registration result using the standard ICP approach. Note that the blue view is wrongly moved between the two red structures. Conversely, when X84-ICP is used, the red structure on the left is recognized as outliers and the views are correctly aligned, see Fig. 8.19 (right).

We highlight that although X84-ICP performs well in these experiments, in more general cases if the number of outliers is greater than 50% of the residual distribution the X84 rule is likely to fail.

8.6 Case Study 2: ICP with Levenberg–Marquardt

In this section, we describe a registration method called Levenberg–Marquardt ICP (LM-ICP), which addresses several of the issues of ICP by modeling the registration as a general optimization problem. LM-ICP [54] was proposed in order to minimize the alignment error by employing a nonlinear optimization procedure. The advantage

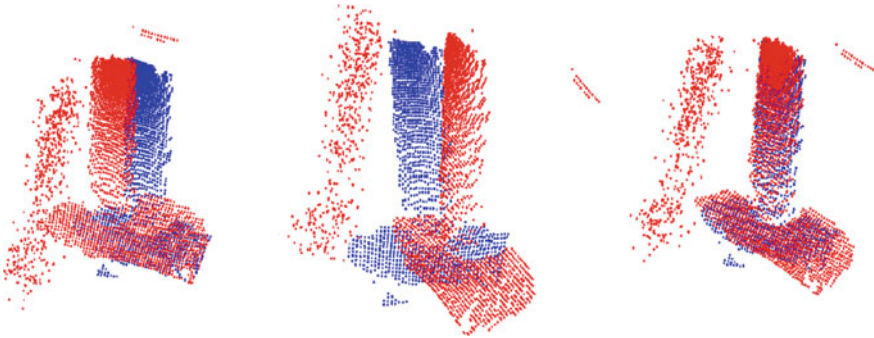


Fig. 8.19 Registration with very noisy images from underwater scenarios composed of tubular structures. Starting views (left), registration using standard ICP, and registration using the robust X84-ICP

of the LM-ICP is the versatility in the definition of the optimization function in order to take into account several aspects of the registration, such as the outlier rejection and the speed.

8.6.1 The LM-ICP Method

The general problem formulation is defined as for the ICP algorithm. The error function $E(\mathbf{a}) = E_{ICP}(\mathbf{a}, \mathbb{D}, \mathbb{M})$ is nonlinear least squares and can thus be written as the sum of N_d squared residual vectors:

$$E(\mathbf{a}) = \sum_{i=1}^{N_d} (e_i(\mathbf{a}))^2, \quad e_i(\mathbf{a}) = \|\mathbf{R}\mathbf{d}_i + \mathbf{t} - \mathbf{m}_j\|. \quad (8.12)$$

Defining the residual vector as

$$\mathbf{e}(\mathbf{a}) = (e_1(\mathbf{a}) \ e_2(\mathbf{a}) \ \cdots \ e_{N_d}(\mathbf{a}))^T, \quad (8.13)$$

we rewrite the error function as $E(\mathbf{a}) = \|\mathbf{e}(\mathbf{a})\|^2$.

The Levenberg–Marquardt algorithm combines the methods of gradient descent and Gauss–Newton. The goal at each iteration is to choose an update to the current estimate \mathbf{a}_k , say \mathbf{x} , so that setting $\mathbf{a}_{k+1} = \mathbf{a}_k + \mathbf{x}$ reduces the registration error.

We first derive the Gauss–Newton update. Expanding $E(\mathbf{a} + \mathbf{x})$ to second order yields

$$E(\mathbf{a} + \mathbf{x}) = E(\mathbf{a}) + (\nabla E(\mathbf{a}) \cdot \mathbf{x}) + \frac{1}{2!}((\nabla^2 E(\mathbf{a}) \cdot \mathbf{x}) \cdot \mathbf{x}) + h.o.t. \quad (8.14)$$

This is rewritten in terms of \mathbf{e} as

$$\begin{aligned} E(\mathbf{a}) &= \mathbf{e}^\top \mathbf{e} \\ \nabla E(\mathbf{a}) &= 2(\nabla \mathbf{e})^\top \mathbf{e} \\ \nabla^2 E(\mathbf{a}) &= 2(\nabla^2 \mathbf{e}) \mathbf{e} + 2(\nabla \mathbf{e})^\top \nabla \mathbf{e}. \end{aligned}$$

We now define the $N_d \times p$ *Jacobian* matrix $\mathbf{J} = \nabla \mathbf{e}$, with block (i, j) as $\mathbf{J}_{i,j} = \frac{\partial E_i}{\partial \mathbf{a}_j}$ (p is the number of elements in \mathbf{a}). Introducing the Gauss–Newton approximation (i.e., neglecting $(\nabla^2 \mathbf{e}) \mathbf{e}$), we get

$$E(\mathbf{a} + \mathbf{x}) \approx \mathbf{e}^\top \mathbf{e} + \mathbf{x}^\top \mathbf{J}^\top \mathbf{e} + \mathbf{x}^\top \mathbf{J}^\top \mathbf{J} \mathbf{x}. \quad (8.15)$$

Differentiating with respect to \mathbf{x} and nullifying yields

$$\nabla_{\mathbf{x}} E(\mathbf{a} + \mathbf{x}) = \mathbf{J}^\top \mathbf{e} + \mathbf{J}^\top \mathbf{J} \mathbf{x} = 0, \quad (8.16)$$

and gives the Gauss–Newton update:

$$\mathbf{x}_{GN} = -(\mathbf{J}^\top \mathbf{J})^{-1} \mathbf{J}^\top \mathbf{e}. \quad (8.17)$$

Gauss–Newton is usually fast for mildly nonlinear problems (it has superlinear convergence speed), but there is no guarantee of convergence in the general case (an update may increase the error).

We now derive the gradient descent update. Since we deal with a least squares problem, the gradient descent update is simply given by

$$\mathbf{x}_{GD} = -\lambda^{-1} \mathbf{J}^\top \mathbf{e}, \quad (8.18)$$

where λ is the inverse step length. Gradient descent has the nice property that, unless a local minimum has been reached, one can always decrease the error by making the step length small enough. On the other hand, gradient descent is known to be slow and rather inefficient.

The Levenberg–Marquardt algorithm combines both Gauss–Newton and gradient descent updates in a relatively simple way:

$$\mathbf{x}_{LM} = -(\mathbf{J}^\top \mathbf{J} + \lambda \mathbf{I})^{-1} \mathbf{J}^\top \mathbf{e}. \quad (8.19)$$

A large value of λ yields a small, safe, gradient descent step while a small value of λ favor large and more accurate steps of Gauss–Newton that make convergence to a local minimum faster. The art of a Levenberg–Marquardt algorithm implementation is in tuning λ after each iteration to ensure rapid progress even where Gauss–Newton fails. The now-standard implementation is to multiply λ by 10 if the error increases and to divide it by 10 if the error decreases (with an upper bound at 10^8 and a lower bound at 10^{-4} , for instance). In order to make the method robust to outliers, one

may attenuate the influence of points with a large error by replacing the square error function by an M-estimator ϵ and an Iterative Reweighted Least Squares (IRLS)-like reweighting procedure. For instance, the following robust functions can be used:

$$\text{Lorenzian: } \epsilon(r) = \log\left(1 + \frac{r^2}{\sigma}\right) \quad \text{or} \quad \text{Huber: } \epsilon(r) = \begin{cases} r^2 & r < \sigma \\ 2\sigma|r| - \sigma^2 & \text{otherwise.} \end{cases}$$

8.6.2 Computing the Derivatives

An important issue in how Levenberg–Marquardt is applied to ICP is the one of computing the derivatives of the error function. The simplest approach is based on using finite differencing, assuming that the error function is smooth. However, this leads to a cost of p extra function evaluations per inner loop. In [54], a more effective solution was proposed based on the *distance transform* which also drastically improves the computational efficiency. The distance transform is defined as

$$D_\epsilon(\mathbf{x}) = \min_j \epsilon^2(\|\mathbf{m}_j - \mathbf{x}\|), \quad (8.20)$$

where $\mathbf{x} \in \mathbb{X}$ and \mathbb{X} is a discrete grid representing the volume which encloses the model-view \mathbb{M} . Indeed, each data point d_i can be easily associated to grid points by obtaining the residual error $e_i = X(d_i)$ in one shot.⁷ In other words, LM-ICP merges the two main steps of ICP, namely, closest point computation and transformation estimation, in a single step. Note further that when the mapping $\|\mathbf{x}\| \rightarrow \epsilon^2(\|\mathbf{x}\|)$ is monotonic, we obtain that $D_\epsilon(\mathbf{x}) = \epsilon^2(\|D(\mathbf{x})\|)$, so existing algorithms to compute D may be used to compute D_ϵ , without requiring knowledge of the form of ϵ .

By combining Eq. (8.12) with Eq. (8.20), the new formulation of the registration problem becomes

$$E(\mathbf{a}) = \sum_{i=1}^{N_d} D_\epsilon(T(\mathbf{a}, \mathbf{d}_i)). \quad (8.21)$$

This formulation makes it much easier to compute the derivatives of E . In fact, since the distance transform is computed in a discrete form, it is possible to compute finite differences derivatives. More specifically, $\nabla_{\mathbf{x}} D_\epsilon = [\frac{\partial D_\epsilon}{\partial x}, \frac{\partial D_\epsilon}{\partial y}, \frac{\partial D_\epsilon}{\partial z}]$ is computed by defining $\frac{\partial D_\epsilon(x, y, z)}{\partial x} = \frac{D_\epsilon(x+1, y, z) - D_\epsilon(x-1, y, z)}{2}$, $\frac{\partial D_\epsilon(x, y, z)}{\partial y} = \frac{D_\epsilon(x, y+1, z) - D_\epsilon(x, y-1, z)}{2}$, and $\frac{\partial D_\epsilon(x, y, z)}{\partial z} = \frac{D_\epsilon(x, y, z+1) - D_\epsilon(x, y, z-1)}{2}$. In practice, $\nabla_{\mathbf{x}} D_\epsilon$ remains constant through the minimization, and we get

⁷Note that the volume is discretized into integer values; therefore, the data point d_i should be rounded to recover $X(d_i)$.

$$\nabla_{\mathbf{a}} E(\mathbf{a}) = \sum_{i=1}^{N_d} \nabla_{\mathbf{x}} D_{\epsilon}(T(\mathbf{a}, \mathbf{d}_i)) \nabla_{\mathbf{a}}^{\top} T(\mathbf{a}, \mathbf{d}_i). \quad (8.22)$$

Note that the computation of $\nabla_{\mathbf{a}}^{\top} T(\mathbf{a}, \mathbf{d}_i)$ depends on the rigid transformation parametrization being used. In [54], the author proposed model rotations by unitary quaternions for which the derivatives can be easily computed analytically. Finally, in order to compute the derivatives using matrix operators, the *Jacobian* matrix is defined as $\mathbf{J}_{i,j} = (\nabla_{\mathbf{x}} D_{\epsilon}(T(\mathbf{a}, \mathbf{d}_i)) \cdot \nabla_{a_j}^{\top} T(\mathbf{a}, \mathbf{d}_i))$, where $\nabla_{a_j} T(\mathbf{a}, \mathbf{d}_i) = [\frac{\partial T_x(\mathbf{a}, \mathbf{d}_i)}{\partial a_j}, \frac{\partial T_y(\mathbf{a}, \mathbf{d}_i)}{\partial a_j}, \frac{\partial T_z(\mathbf{a}, \mathbf{d}_i)}{\partial a_j}]$.

8.6.3 The Case of Quaternions

Let the quaternion be defined by $\mathbf{q} = [s, \mathbf{v}]$ where s and \mathbf{v} are the scalar and vectorial components, respectively [153]. Let \mathbf{d} be the point on which the rotation must be applied. To this aim such a point must be represented in quaternion space, leading to $\mathbf{r} = [0, \mathbf{d}]$. Therefore, the rotated point is obtained by

$$\mathbf{r}' = \mathbf{q} \mathbf{r} \mathbf{q}^{-1}.$$

By multiplying in quaternion space,⁸ we obtain

$$\mathbf{r}' = [0, s^2 \mathbf{d} + (\mathbf{d} \cdot \mathbf{v}) \cdot \mathbf{v} + 2s(\mathbf{v} \times \mathbf{d}) + \mathbf{v} \times (\mathbf{v} \times \mathbf{d})].$$

We represent this rotated point as

$$\mathbf{r}' = [0, T_x, T_y, T_z],$$

where

$$\begin{aligned} T_x &= s^2 d_x + (d_x v_x + d_y v_y + d_z v_z) v_x + 2s(v_y d_z - v_z d_y) + v_y(v_x d_y - v_y d_x) - v_z(v_z d_x - v_x d_z) = \\ &= s^2 d_x + v_x^2 d_x + v_x v_y d_y + v_x v_z d_z + 2s v_y d_z - 2s v_z d_y + v_x v_y d_y - v_y^2 d_x - v_z^2 d_x + v_x v_z d_z = \\ &= (s^2 + v_x^2 - v_y^2 - v_z^2) d_x + 2(v_x v_y - s v_z) d_y + 2(v_x v_z + s v_y) d_z \end{aligned}$$

$$\begin{aligned} T_y &= s^2 d_y + (d_x v_x + d_y v_y + d_z v_z) v_y + 2s(v_z d_x - v_x d_z) + v_z(v_y d_z - v_z d_y) - v_x(v_x d_y - v_y d_x) = \\ &= s^2 d_y + v_x v_y d_x + v_y^2 d_y + v_y v_z d_z + 2s v_z d_x - 2s v_x d_z + v_y v_z d_z - v_z^2 d_y - v_x^2 d_y + v_x v_y d_x = \\ &= 2(v_x v_y + s v_z) d_x + (s^2 - v_x^2 + v_y^2 - v_z^2) d_y + 2(v_y v_z - s v_x) d_z \end{aligned}$$

⁸A multiplication between two quaternions \mathbf{q} and \mathbf{q}' is defined as $[ss' - \mathbf{v} \cdot \mathbf{v}', \mathbf{v} \times \mathbf{v}' + s\mathbf{v}' + s'\mathbf{v}]$.

$$\begin{aligned}
T_z &= s^2 d_z + (d_x v_x + d_y v_y + d_z v_z) v_z + 2s(v_x d_y - v_y d_x) + v_x(v_z d_x - v_z d_z) - v_y(v_y d_z - v_z d_y) = \\
&= s^2 d_z + v_x v_z d_x + v_y v_z d_y + v_z^2 d_z + 2s v_x d_y - 2s v_y d_x + v_x v_z d_x - v_x^2 d_z - v_y^2 d_z + v_y v_z d_y = \\
&= 2(v_x v_z - s v_y) d_x + 2(v_y v_z + s v_x) d_y + (s^2 - v_x^2 - v_y^2 + v_z^2) d_z.
\end{aligned}$$

Now we introduce the translation component $[t_x, t_y, t_z]$ and normalize the quaternion by obtaining

$$\begin{aligned}
T_x &= \frac{(s^2 + v_x^2 - v_y^2 - v_z^2) d_x}{s^2 + v_x^2 + v_y^2 + v_z^2} + \frac{2(v_x v_y - s v_z) d_y}{s^2 + v_x^2 + v_y^2 + v_z^2} + \frac{2(v_x v_z + s v_y) d_z}{s^2 + v_x^2 + v_y^2 + v_z^2} + t_x \\
T_y &= \frac{2(v_x v_y + s v_z) d_x}{s^2 + v_x^2 + v_y^2 + v_z^2} + \frac{(s^2 - v_x^2 + v_y^2 - v_z^2) d_y}{s^2 + v_x^2 + v_y^2 + v_z^2} + \frac{2(v_y v_z - s v_x) d_z}{s^2 + v_x^2 + v_y^2 + v_z^2} + t_y \\
T_z &= \frac{2(v_x v_z - s v_y) d_x}{s^2 + v_x^2 + v_y^2 + v_z^2} + \frac{2(v_y v_z + s v_x) d_y}{s^2 + v_x^2 + v_y^2 + v_z^2} + \frac{(s^2 - v_x^2 - v_y^2 + v_z^2) d_z}{s^2 + v_x^2 + v_y^2 + v_z^2} + t_z.
\end{aligned}$$

According to this model for rotation and translation, the vector of unknowns is $\mathbf{a} = [s, v_x, v_y, v_z, t_x, t_y, t_z]$ (i.e., $\mathbf{a} \in \mathbb{R}^7$). Therefore, the Jacobian part $\nabla_{\mathbf{a}}^T T(\mathbf{a}, \mathbf{d})$ is a 3×7 matrix:

$$\nabla_{\mathbf{a}}^T T(\mathbf{a}, \mathbf{d}) = \begin{pmatrix} \frac{\partial T_x}{\partial s} & \frac{\partial T_x}{\partial v_x} & \frac{\partial T_x}{\partial v_y} & \frac{\partial T_x}{\partial v_z} & \frac{\partial T_x}{\partial t_x} & \frac{\partial T_x}{\partial t_y} & \frac{\partial T_x}{\partial t_z} \\ \frac{\partial T_y}{\partial s} & \frac{\partial T_y}{\partial v_x} & \frac{\partial T_y}{\partial v_y} & \frac{\partial T_y}{\partial v_z} & \frac{\partial T_y}{\partial t_x} & \frac{\partial T_y}{\partial t_y} & \frac{\partial T_y}{\partial t_z} \\ \frac{\partial T_z}{\partial s} & \frac{\partial T_z}{\partial v_x} & \frac{\partial T_z}{\partial v_y} & \frac{\partial T_z}{\partial v_z} & \frac{\partial T_z}{\partial t_x} & \frac{\partial T_z}{\partial t_y} & \frac{\partial T_z}{\partial t_z} \end{pmatrix}, \quad (8.23)$$

where T_x, T_y , and T_z have been defined above. For instance, we can compute the derivative component $\frac{\partial T_x}{\partial v_x}$ as

$$\begin{aligned}
\frac{\partial T_x}{\partial v_x} &= \frac{2v_x d_x}{s^2 + v_x^2 + v_y^2 + v_z^2} - \frac{2v_x(s^2 + v_x^2 - v_y^2 - v_z^2) d_x}{(s^2 + v_x^2 + v_y^2 + v_z^2)^2} + \\
&+ \frac{2v_x d_y}{s^2 + v_x^2 + v_y^2 + v_z^2} - \frac{4v_x(v_x v_y - s v_z) d_y}{(s^2 + v_x^2 + v_y^2 + v_z^2)^2} + \\
&+ \frac{2v_x d_z}{s^2 + v_x^2 + v_y^2 + v_z^2} - \frac{4v_x(v_x v_z + s v_y) d_z}{(s^2 + v_x^2 + v_y^2 + v_z^2)^2}.
\end{aligned}$$

Similarly, all the other components of the Jacobian can easily be computed.

8.6.4 Summary of the LM-ICP Algorithm

The algorithm for LM-ICP can be summarized as

1. Set $\lambda \leftarrow \lambda_0 = 10$,

2. compute distance transform $D_\epsilon(\mathbf{x})$,
3. set $\mathbf{a}_k \leftarrow \mathbf{a}_0$,
4. compute $\mathbf{e}_k = \mathbf{e}(\mathbf{a}_k)$,
5. compute \mathbf{J} ,
6. repeat
 - a. compute update $\mathbf{a}_{k+1} = \mathbf{a}_k - (\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I})^{-1} \mathbf{J}^T \mathbf{e}_k$ ⁹
 - b. compute $\Delta E = E(\mathbf{a}_{k+1}) - E(\mathbf{a}_k)$
 - c. If $\Delta E > 0$ then $\lambda = 10\lambda$, go to a, else $\lambda = \frac{1}{10}\lambda$, go to 4.
7. If $\|\mathbf{e}_k\| > \nu$ go to 3, else terminate.

Note that ν is a constant which defines the convergence of the algorithm. As already highlighted, the algorithm above is the standard LM algorithm. The crucial components are (i) the choice of unknowns \mathbf{a} , (ii) the computation of error vector \mathbf{e} , and (iii) the computation of the Jacobian matrix \mathbf{J} . In particular, the distance transform $D_\epsilon(\mathbf{x})$ enables an improvement in the computational efficiency of the error computation and makes the computation of the Jacobian feasible. The starting value \mathbf{a}_0 can be estimated by employing some of the techniques described in Sect. 8.2.3.1.

8.6.5 Results and Discussion

Figure 8.20 shows an example of LM-ICP alignment between two views. In this experiment, the emphasis is on the speed of the algorithm, since the accuracy is guaranteed by the fact that the two views are well overlapped. The LM-ICP takes less than 1 s for an LM iteration. A total of 20 iterations has been run to reach convergence. Both the data-view and the model-view have about 40,000 points. Using the basic *ICP* algorithm, the same number of iterations are required but each iteration takes more than 30 s. This confirms that a drastic improvement in speed is observed with LM-ICP, in comparison with basic ICP. Note that a crucial parameter is the grid size. It trades off computational efficiency with memory space. Moreover, it requires that the data scan is always inside the volume by requiring large memory space for storage when only a small overlap is observed between the views. Further experiments can be found in [54]. More details on experimental setup can be found on the LM-ICP website.¹⁰ In practice, LM-ICP also enlarges the basin of convergence and estimates a more accurate solution (the minimum is reached with 50% fewer iterations on average, see [54] for more details).

Finally, it is worth noting that LM-ICP can be easily extended to apply many other variants of the ICP. Multiview registration could also be solved in the LM-ICP framework.

⁹While we have chosen the identity as the damping matrix, some authors rather choose the diagonal part of the Gauss–Newton Hessian approximation.

¹⁰<http://research.microsoft.com/en-us/um/people/awf/lmicp>.

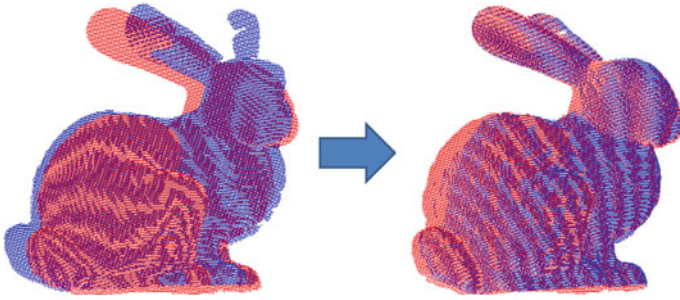


Fig. 8.20 LM-ICP. The starting pose (left) and merged views after registration (right)

8.7 Case Study 3: Deformable ICP with Levenberg–Marquardt

In this section, we describe an advanced registration technique: *Deformable-Levenberg Marquardt Iterative Closest Point (DLM-ICP)* [31]. DLM-ICP extends the LM-ICP approach, introduced in Sect. 8.6, to deformable objects. We focus on continuous smooth surfaces such as the page of a book being turned in front of a range sensor. To this aim, a *template* model is warped toward the input scans in order to capture surface deformations. In this case, several instances of almost the entire time-varying object are observed rather than different points of view of an object, and the aim of registration is to align the views over time using a registration-by-fitting approach.

The template model introduces a prior on the acquired shape by providing a joint registration and reconstruction of the object with hole-filling and noise removal. The proposed method exploits only geometric information without the extraction of feature points. According to [54], described in Sect. 8.6, registration is modeled as an optimization problem defined by an error function whose global minimum is the sought after solution, estimated by the Levenberg–Marquardt algorithm. The error function introduces the constraint that data points must be close to model points (i.e., the template). As for [54], it explicitly embeds a min operator, thus avoiding the traditional two steps in ICP-like algorithms, through the use of a distance transform. Furthermore, thanks to the flexibility of LM, many other terms are introduced to model different expected behaviors of the deformation, namely, *surface*, and *temporal smoothness* as well as *inextensibility* of the surface. Finally, a boundary constraint is introduced to prevent the computed surface from sliding arbitrarily.

We highlight that, with this method, the unknowns are the template model, represented by a planar mesh that is deformed to fit each point cloud. More specifically, we directly estimate the position of the model points without imposing any prior about the kind of transformation function that has been applied. In particular, each unknown (i.e., each vertex of the template) influences a very small portion of the error function. Indeed, another interesting property of DLM-ICP is that the Jacobian

matrix, involved in the normal equations to be solved at each iteration, is highly sparse for all the terms. This makes the estimation of dense deformation fields tractable and fast.

8.7.1 Surface Representation

The sequence of 3D point clouds D_i , with $N_d = l_i$ points each, is represented by

$$D_i = \begin{pmatrix} d_{i,1}^x & d_{i,1}^y & d_{i,1}^z \\ \vdots & \vdots & \vdots \\ d_{i,l_i}^x & d_{i,l_i}^y & d_{i,l_i}^z \end{pmatrix}.$$

The unknown model, $\mathbf{a} = \mathbf{M}$, has a grid structure and is thus represented by three $R \times C$ matrices, giving the grid's deformation. Each matrix is reshaped in a single vector of size $N_m = RC$, giving \mathbf{M}_i as

$$\mathbf{M}_i = \begin{pmatrix} m_{i,1}^x & m_{i,1}^y & m_{i,1}^z \\ \vdots & \vdots & \vdots \\ m_{i,N_m}^x & m_{i,N_m}^y & m_{i,N_m}^z \end{pmatrix}.$$

In practice, the number of data points is much larger than the number of model points (i.e., $l_i \mathbf{g} N_m$). Upon convergence, the algorithm determines, for each model point, if there is a corresponding point in the current point cloud. Points may be missing because of occlusions or corrupted sensor output. This approach has the advantage that it naturally gives the reconstructed surface by interpolating the mesh points. Point cloud registration is obtained by composing the deformation fields. Note that, in contrast to Sect. 8.6, the registration is from model points to data points.

8.7.2 Cost Function

The cost function combines two *data* and three *penalty* terms:

$$E(\mathbf{M}) = E_g(\mathbf{M}) + \lambda_b E_b(\mathbf{M}) + \lambda_s E_s(\mathbf{M}) + \lambda_t E_t(\mathbf{M}) + \lambda_x E_x(\mathbf{M}), \quad (8.24)$$

where λ_b , λ_s , λ_x , and λ_t are weights. Note that we drop the frame index i for purposes of clarity, and denote \mathbf{M}_i as \mathbf{M} and \mathbf{M}_{i-1} as $\tilde{\mathbf{M}}$.

The data terms are used to attract the estimated surface to the actual point cloud. The first term E_g is for global attraction, while the second one E_b deals with the boundary. In particular, the boundary term aims at preserving the method against

possible sliding of the model along the observed surface. Moreover, these terms must account for possible erroneous points by using robust statistics. The penalty terms are E_s , E_t , and E_x . The first two account for *spatial smoothness* and *temporal smoothness* E_s , respectively. The third one penalizes the *surface stress* and is related to the non-extensibility of the surface, and therefore to material properties of the surface.

This cost function is minimized in an ICP-like manner, as described in the previous section. All five terms are explained below in detail.

Data Term: Global Surface Attraction

This term globally attracts the model to the data points in a closest point manner [126]. Denoting \mathbb{B}_M as the set of boundary points in the model, \mathbb{M} , where

$$\mathbb{M} = \{(m_i^x \ m_i^y \ m_i^z)^T\}, \quad i = 1 \dots N_m \quad (8.25)$$

and \mathbb{B}_D as the set of boundary points in the data, \mathbb{D} , where

$$\mathbb{D} = \{(d_i^x \ d_i^y \ d_i^z)^T\}, \quad i = 1 \dots l_i \quad (8.26)$$

we get the following data term, integrating the model to data points matching step:

$$\sum_{\mathbf{m} \in \mathbb{M} \setminus \mathbb{B}_M} \min_{\mathbf{d} \in \mathbb{D} \setminus \mathbb{B}_D} \|\mathbf{d} - \mathbf{m}\|^2, \quad (8.27)$$

where \mathbf{d} and \mathbf{m} are 3-vectors representing a data point and a model point, respectively. As we mentioned before, the unknowns are not the rigid transformation parameters (i.e., the classical rotation–translation) but correspond to the whole *deformable motion field* in \mathbf{M} .

An *outlier rejection* strategy is introduced by defining a robust function ϵ . Here, the X84 rule is employed [29]. Therefore, Eq. 8.27 is modified so as to get the following robustified data term:

$$E_g(\mathbf{M}) = \sum_{\mathbf{m} \in \mathbb{M} \setminus \mathbb{B}_M} \epsilon \left(\min_{\mathbf{d} \in \mathbb{D} \setminus \mathbb{B}_D} \|\mathbf{d} - \mathbf{m}\|^2 \right). \quad (8.28)$$

Data Term: Boundary Attraction

This term attracts boundary model points to boundary data points. It is defined in a similar manner to the global attraction term (Eq. 8.28) except that the sum and min operators are over the boundary points:

$$E_b(\mathbf{M}) = \sum_{\mathbf{m} \in \mathbb{B}_M} \epsilon \left(\min_{\mathbf{d} \in \mathbb{B}_D} \|\mathbf{d} - \mathbf{m}\|^2 \right). \quad (8.29)$$

Note that the boundaries can be computed by combining edge detection techniques with morphological operators.¹¹ More precisely, from the range image, we detect the portion of the image which is covered by the object we want to track (i.e., a piece of paper), and we impose the condition that boundaries of the model and the observed surface must coincide.

Penalty Term: Spatial Smoothness

This term discourages surface discontinuities by penalizing its second derivatives, as an approximation to its curvature. According to the definition of the geometry image [62], the model \mathbb{M} is a displacement field parameterized¹² by (u, v) with $u = [1 \dots R]$ and $v = [1 \dots C]$, i.e., $\mathbf{M}(u, v) = (M^x(u, v) M^y(u, v) M^z(u, v))^T$. The spatial smoothness term can thus be taken as the surface bending energy:

$$E_s(\mathbf{M}) = \int \int \left\| \frac{\partial \mathbb{M}^2}{\partial^2 u} \right\|^2 + 2 \left\| \frac{\partial \mathbb{M}^2}{\partial u \partial v} \right\|^2 + \left\| \frac{\partial \mathbb{M}^2}{\partial^2 v} \right\|^2 du dv.$$

Using a finite difference approximation for the first and second derivatives [116], the bending energy can be expressed in discrete form as a quadratic function of \mathbb{M} . More specifically, the derivative $\frac{\partial \mathbb{M}^x}{\partial u}$ at a point (u, v) is discretely approximated as $\frac{\partial \mathbb{M}^x(u,v)}{\partial u} = M^x(u+1, v) - M^x(u-1, v)$. This can be conveniently represented by a constant $N_m \times N_m$ matrix \mathbf{C}_u such that $\nabla_u \mathbb{M}^x = \mathbf{C}_u \cdot \text{vect}(\mathbf{M}^x)$, where $\text{vect}(\mathbf{M}^x)$ is the vectorization operator which rearranges matrix \mathbf{M}^x to a vector. A similar matrix \mathbf{C}_v can be computed with respect to v . Indeed, the second derivatives are computed using Hessian operator matrices, namely, \mathbf{C}_{uu} , \mathbf{C}_{uv} , \mathbf{C}_{vv} . The surface bending energy can be expressed in discrete form by defining

$$E_s^x = \text{vect}(\mathbf{M}^x)^T (\mathbf{C}_{uu}^T \mathbf{C}_{uu} + 2\mathbf{C}_{uv}^T \mathbf{C}_{uv} + \mathbf{C}_{vv}^T \mathbf{C}_{vv}) \text{vect}(\mathbf{M}^x),$$

and by computing

$$E_s(\mathbf{M}) = E_s^x(\mathbf{M}^x) + E_s^y(\mathbf{M}^y) + E_s^z(\mathbf{M}^z),$$

which can be further expressed in matrix form as follows:

$$E_s(\mathbf{M}) = \text{vect}(\mathbf{M})^T \mathcal{K} \text{vect}(\mathbf{M}), \quad (8.30)$$

where \mathcal{K} is a $3N_m \times 3N_m$, highly sparse matrix.

Penalty Term: Temporal Smoothness

This term defines a dependency between the current and the previous point clouds, \mathbf{M} and $\tilde{\mathbf{M}}$:

¹¹The object boundaries can be estimated according to the kind of sensor being used. For instance, boundaries on range scans can be estimated on the range image. In stereo sensors, they can be estimated on one of the two optical views.

¹²Recall that the model points lie on a grid.

$$E_t(\mathbf{M}) = \|\mathbf{M} - \tilde{\mathbf{M}}\|^2. \quad (8.31)$$

This makes the surface deformation smooth over time and can be used within a sequential processing approach. Obviously, it is not used in the first frame of the sequence.

Penalty Term: Non-extensibility

This term discourages surface stretching. It encourages mesh vertices to preserve their distance with their local neighborhood [129]:

$$E_x(\mathbf{M}) = \sum_{\mathbf{m} \in \mathbb{M}} \sum_{\mathbf{k} \in \mathcal{N}(\mathbf{m})} (\|\mathbf{m} - \mathbf{k}\|^2 - L_{m,k}^2)^2, \quad (8.32)$$

where $L_{m,k}$ are constants, which are computed at the first frame after robust initialization, and $\mathcal{N}(\mathbf{m})$ is the 8-neighborhood of the mesh vertex \mathbf{m} .

8.7.3 Minimization Procedure

The DLM-ICP cost function (8.24) is a sum of squared residuals, nonlinearly depending on the unknowns in \mathbf{M} . Therefore, as in Sect. 8.6, the Levenberg–Marquardt algorithm can be used. In order to provide partial derivatives of the residuals through a Jacobian matrix, all five terms in the cost function are separately differentiated and stacked as

$$\mathbf{J}^\top = (\mathbf{J}_d^\top \quad \mathbf{J}_b^\top \quad \mathbf{J}_s^\top \quad \mathbf{J}_t^\top \quad \mathbf{J}_x^\top), \quad (8.33)$$

where $\mathbf{J}_d^{N_m \times 3N_m}$, $\mathbf{J}_b^{N_B \times 3N_m}$, $\mathbf{J}_s^{3N_m \times 3N_m}$, $\mathbf{J}_t^{N_m \times 3N_m}$, and $\mathbf{J}_x^{\xi \times 3N_m}$ are related to the global attraction, boundary attraction, spatial smoothness, temporal smoothness, and non-extensibility terms, respectively, and $\xi = \text{size}(\mathcal{N}(\mathcal{M}))$. In particular, the Jacobians of global and boundary attraction terms are estimated by finite differences through distance transform, as described in Sect. 8.6.

Note that, in this case, since the Hessian matrix¹³ $\mathbf{H} = \mathbf{J}^\top \mathbf{J} + \lambda \mathbf{I}$ must be inverted at each LM iteration, the problem is not tractable if the number of model points is too high (if the deformation field is too dense). One advantage of the proposed approach is that the Jacobian matrix \mathbf{J} is very sparse. Thus, it uses the sparsity to speed up each iteration using the technique in [114]. In particular, a sparse Cholesky factorization package can be used, as in the Matlab “mldivide” function.

¹³We use “Hessian matrix” for the damped Gauss–Newton approximation to the true Hessian matrix.

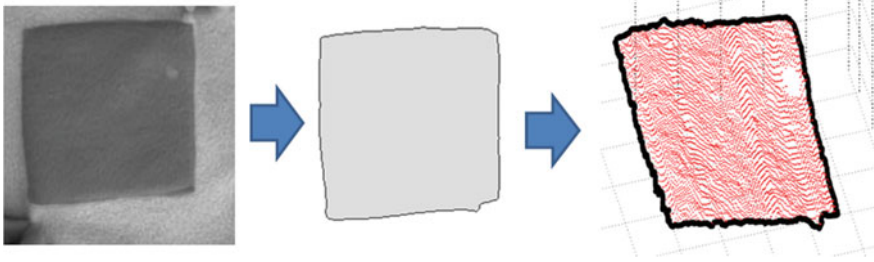


Fig. 8.21 Data acquisition: intensity image of the blanket (left), image boundary (center), and the 3D point cloud (right)

8.7.4 Summary of the Algorithm

The DLM-ICP algorithm can be summarized as follows:

1. Choose the model size $R \times C$ (for instance, 10×10)
2. Initialize the template model M_0
3. For each data frame \mathbb{D}_i
 - a. Extract data boundary \mathbb{B}_D
 - b. Set $M_i = M_{i-1}$ to initialize the LM algorithm
 - c. Apply LM-ICP to estimate M_i by minimizing the error function
 - d. Go to 3.

Step 3.c is described in Sect. 8.6.4. Here, the unknown is $\mathbf{a} = M_i$, the error function $E(M_i)$ is defined by Eq. (8.24), and the Jacobian J is defined by Eq. (8.33).

8.7.5 Experiments

In the following experiment, the sensor is a real-time passive-stereo system.¹⁴ The sensor acquires images at 25 FPS (frames-per-second) and provides both intensity (i.e., 2D) and 3D information. The deformation of a portion of a blanket is modeled. Figure 8.21 shows a picture of the blanket. Intensity information is used to segment the boundary; more precisely, only the portion delimited by the dark square is considered. Figure 8.21 also shows the image boundary extracted by combining a binary image segmentation method with 2D morphological operators and depicts the 3D data (i.e., the selected point cloud and 3D boundary).

The sequence is made of 100 point clouds. A model of size $R = 15$ and $C = 20$ is used. Model initialization M_0 is carried out by lying the model grid on a plane which is fitted to the extracted point cloud. Model initialization is employed in the first frame only. Then, each iteration uses the output of the previous one as an initial condition.

¹⁴Data courtesy of eVS (<http://www.evsys.net>).

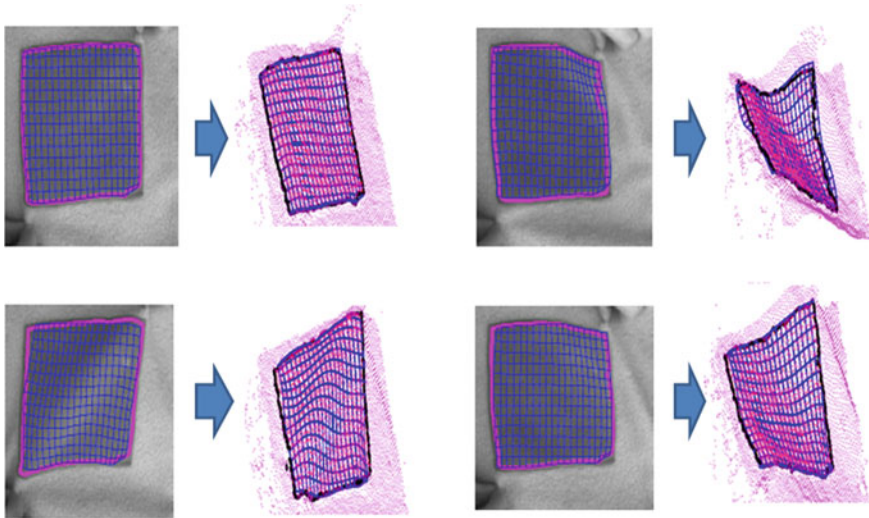


Fig. 8.22 Blanket sequence: 4 selected frames. For each frame, the 2D intensity and the 3D data are visualized. The grid models are shown in 3D space, as well as their projection in the 2D image

Note that a higher value of λ_b is necessary (i.e., $\lambda_b = 1.5$) for a correct convergence of the algorithm to the optimal solution. The other terms are set almost equally to 1. The distance transform parameters are important: the size of the voxels trades off speed and accuracy. In this experiment, the volume is divided into $36 \times 36 \times 18$ voxels. Figure 8.22 shows a selection of the output sequence. For each frame, we visualize (i) the intensity image with the extracted 2D boundary and the 2D projection of the estimated model and (ii) the point cloud, after the region-of-interest selection, evidencing both the 3D boundary and the grid. The blanket is handled from the bottom-left and upper-right corners, respectively. On the early frames, the blanket is gradually bent toward the square center; then, it is strongly stretched, moving the corners far from each other. Finally, in the late frames, random deformations are generated, especially around the corners. Results are satisfying since the fitting is correct for the whole sequence, in spite of the presence of strong occlusions and deformations. The mesh grids are well superimposed on data points maintaining a smooth shape. Nevertheless, the projection of the grids to the 2D images confirm the accuracy of the registration. More details on performance evaluation are available in [31].

8.8 Case Study 4: Computer-Aided Laparoscopy by Preoperative Data Registration

We describe a use case of 3D registration in laparoscopy. The technique involves 3D–3D deformable registration solved by customizing DLM-ICP.

8.8.1 Context

In laparoscopy, the surgeon uses keyholes in the patient’s abdominal wall. The laparoscope is the observation device and consists of a camera connected to a thin rod containing an optic fiber. The surgeon typically uses between 2–4 keyholes of about 1 cm in diameter. Laparoscopy has many advantages over open surgery. However, some elements of the anatomy may be difficult to locate. This is because the organs and tissues are generally opaque. Structures such as the internal tumors and vessels are therefore invisible. Registration techniques can be used to circumvent this limitation of visualization by enabling intraoperative augmented reality. The most promising approach is to use preoperative data such as a Magnetic Resonance (MR) or Computed Tomography (CT) scan. These modalities are generally not available during surgery. They show the tumors and vessels well and can be used to create a 3D model of the target organ’s outer surface and internal structures before surgery. The challenge is then to register this preoperative 3D model to the intraoperative 2D images given by the laparoscope. This is tremendously difficult because of the organ’s deformation (due, for instance, to insufflation of the abdominal cavity) and because the vast majority of laparoscopes are monocular. Registration in laparoscopy is still an open problem but promising preliminary approaches have been proposed. We discuss the image-based approach, which uses the image contents to solve registration without requiring the use of fiducials. An example taken from [39] is shown in Fig. 8.23. This example represents the result of using a computer-aided laparoscopy system in a real surgery.

8.8.2 Problem Statement

Augmented reality laparoscopy follows three steps: preoperative 3D model reconstruction, intraoperative registration, and visualization. A complete pipeline is shown in Fig. 8.24. We here focus on the registration step but the visualization step is also highly challenging and researched, see, for instance, [6, 7, 12, 55, 107, 152]. The preoperative 3D model reconstruction consists in segmenting the preoperative 3D volume and interpolating the voxels to create a surface represented by a mesh. This may be extremely challenging to solve automatically but semi-automatic methods

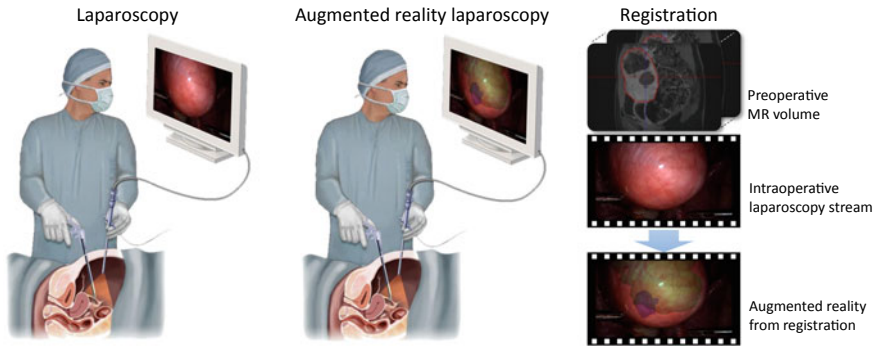


Fig. 8.23 Laparoscopy and the problem of registration in augmented reality laparoscopy. (left) The principle of laparoscopy, here with an example of a myomectomy procedure where the uterus contains two inner and thus invisible tumors. (middle) The principle of augmented reality laparoscopy, where the tumors are shown using virtual transparency. (right) The registration problem to be solved

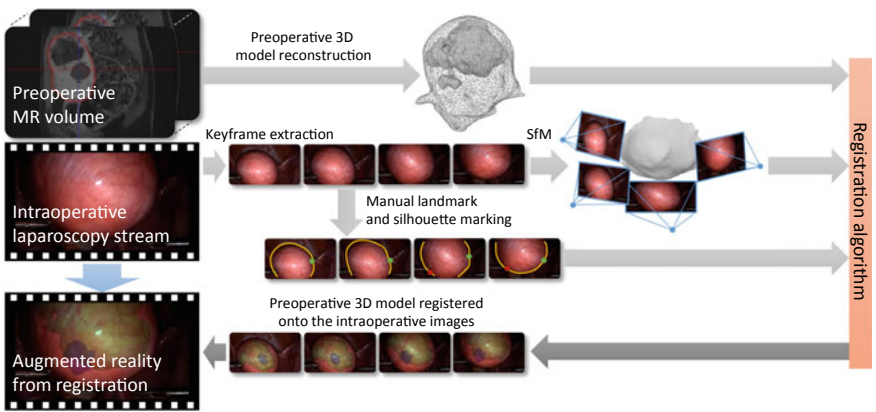


Fig. 8.24 Principle of an ICP-based solution to the problem of registration in augmented reality laparoscopy. The original problem is transformed into a special type of 3D surface-to-surface registration problem, which is solved using a particular instance of deformable ICP [39]

already exist which are very effective and available in software packages such as MITK (the Medical Imaging Interaction Toolkit ¹⁵).

It is fundamental to understand that augmented reality laparoscopy involves two types of registration problem. The first and most complex type occurs at the start of surgery, when the preoperative 3D model has not been related at all to the 2D images yet. It implies that the preoperative 3D model is purely geometric, in other words untextured, as the preoperative data do not contain color, at least not color as we see it in laparoscopy. The second type of registration problem occurs in a later stage of surgery, after the preoperative 3D model has been texture mapped, thanks to the

¹⁵ www.mitk.org.

first round of registration. This is a considerably easier problem as the availability of a texture map makes it possible to find keypoint correspondences between the preoperative 3D model and the 2D images, as, for instance, in [40]. We here focus on the first type of registration problem, which requires an ICP-like approach to draw correspondences and solve for deformation simultaneously.

The registration problem takes as inputs the preoperative 3D model and a set of intraoperative 2D images, extracted from the laparoscopy video stream and produces as output a 3D deformation field which, when applied to the preoperative 3D model, brings it to the state in which the organ was observed during surgery. We make the strong assumption that the organ does not deform across the 2D images. This is a valid assumption at the early steps of surgery, before the organ is cut through. This assumption means that the intraoperative state of the organ's outer part can be recovered by existing techniques such as Structure from Motion (SfM). The case of non-rigidly-related 2D images can be handled in two ways. First, the images can be dealt with individually, as was attempted for liver laparoscopic augmented reality [1, 79]. Second, advanced techniques extending SfM to deformable structures could be used, namely, Non-Rigid Structure from Motion (NRSfM) [19, 108, 127], but their applicability to laparoscopy data has not yet been demonstrated.

8.8.3 Registration

The registration has two main steps. The first step is to compute an intraoperative 3D reconstruction of the organ's outer surface from the 2D images. This is solved automatically by using SfM [39] or Simultaneous Localization and Mapping (SLAM) [57]. This produces a point cloud representing the organ's outer surface and the laparoscope's intrinsic and extrinsic parameters for each 2D image. The second step is to compute the registration for the organ's outer surface, which boils down to a 3D surface-to-surface registration. This can be solved by an existing technique such as DLM-ICP, the deformable ICP presented in the previous section. There is, however, a key difference between DLM-ICP and the case at hand: an organ has a spherical rather than a disk topology and therefore does not have boundaries, as opposed to a piece of paper or cloth. This means that the boundary term E_b from the cost (8.24) of DLM-ICP cannot be used. This second step is thus the most difficult: without the boundary term and as many organs tend to have smooth parts and the intraoperative 3D reconstruction tends to be partial, the 3D registration may easily drift to a false solution as the two surfaces can slide onto one another without changing the registration cost much. An equivalent of the boundary for spherical objects is the silhouette, which may be introduced as a new term in the cost, given that the organ's silhouette has been manually marked in some of the 2D images. The silhouette is, however, much weaker than the boundary term as it helps registration but does not prevent surface sliding. In practice, this effect is mitigated by using a few manually defined anatomical landmarks, depending on the organ at hand. For the uterus, for instance, these may be chosen as the junction points between the Fal-

lopidian tubes and the uterus' body. The general cost function is thus similar to, but not exactly like, the cost (8.24) of DLM-ICP. It combines three data terms and two penalty terms:

$$E(M) = \lambda_{dg}E_{dg}(M) + \lambda_{ds}E_{ds}(M) + \lambda_{dl}E_{dl}(M) + \lambda_{ps}E_{ps}(M) + \lambda_{px}E_{px}(M),$$

where $\lambda_{dg}, \lambda_{ds}, \lambda_{dl}, \lambda_{ps}, \lambda_{px} \in [0, 1]$ are weights.

Recall that the data terms attract the transformed preoperative 3D model to the intraoperative point cloud. The first data term E_{dg} is the global attraction term E_g of DLM-ICP. The second data term E_{ds} measures the distance on the organ's predicted silhouette. The third data term E_{dl} encapsulates the anatomical landmarks. The penalty terms convey prior knowledge on the organ's admissible deformations. The two penalty terms E_{ps} and E_{px} are, respectively, the spatial smoothness and surface stress terms E_s and E_x of DLM-ICP.

Once the organ's outer surface registration has been computed, a final step is to interpolate the surface deformation field to the desired 3D deformation field. This is desired because the registration found from the organ's outer surface is meant to bring the organ's inner structures, which do not belong to the organ's surface, from the preoperative 3D model to the intraoperative 2D images. This interpolation may be solved by means of a simple 3D spline interpolant estimated using the surface mesh's vertices as control points.

8.8.4 Validation

The accuracy of registration is obviously tremendously important in computer-aided laparoscopy, especially in the predicted position of tumors. It is, however, highly challenging to obtain data with ground truth, as by definition this information is not available during surgery. Quantitative validation is thus usually obtained from phantom or ex-vivo models. The above-described system was validated using synthetic tumors introduced in ex-vivo pig kidneys [35]. In this study, 33 tumors were resected by a control group, using standard laparoscopy, and 29 were resected by an AR group, using the computer-aided laparoscopy system. The resected tumors were analyzed a posteriori to see if the margins were negative (the tumor is entirely removed) or positive (the tumors were only partly removed). In the control group, 42.4% of the tumors were either completely missed or had positive margins. In the AR group, only 13.8% of the tumors had positive margins and none was completely missed. This shows that the registration system is sufficiently accurate to significantly improve the surgeon's performance at tumors localization and resection.

8.9 Challenges and Future Directions

Although the recent methods have brought important advances in 3D registration, especially in the challenging scenarios reported in Sects. 8.3–8.8, there are still many open issues to be addressed that make this research field still exciting and promising. New challenges arise from the rapid technological advances of new 3D acquisition systems. Even if recent deep learning methods have successfully exploited the possibility to recover 3D information using only 2D image, there are still convincing reasons to rely on RGBD sensors. Depth data naturally provides global positioning, and not just local 3D pose information or 2D bounding box information in the image space. Further, RGBD information helps one to solve the real-time reconstruction of entire scenes. At the device level, the new generation of depth sensors is now easily available on mobile phones and is more affordable for generic consumers.¹⁶ Moreover, very often depth sensors are integrated on more complex systems for new emerging applications such as automotive, robotic surgery, or forensic tasks. This leads to an explosion of the available data (i.e., big data) that poses new problems for their manipulation. For instance, the integration between depth information acquired from such different acquisition sources yields new issues in dealing with very heterogeneous data characterized by different resolutions, levels of noise, scales, and so on. Moreover, depth sensors can be accompanied by other devices for the acquisition of other kind of information such as infrared data, GPS, Digital Elevation Model (DTM), or MRI scans (for the medical domain). Therefore, new advanced methods need to be studied for the fusion between information of different nature exploiting multimodal registration techniques.

New problems are emerging for the registration of very large-scale scenarios. Indeed, nowadays it is possible to reconstruct not only a scene composed of several buildings but an entire city. For this task, it is important to improve the methods for the matching at the view level by effectively combining features from both the 2D and 3D domains. In particular, it is important to well organize the previously stored information by introducing new indexing methods to combine shape retrieval strategy with registration techniques. Moreover, especially when the large-scale reconstruction is required for entertainment (i.e., video games or movies), a drastic reduction of the memory demand can be obtained by integrating the registration methods for the generation of procedural models (i.e., inverse procedural model).

In the context of deformable registration, the new issues regard the possibility to relax the prior information on the subject to be registered. Rather than working with a well-delimited class such as the human body or the human face, it is interesting to focus on subject with a more generic shape. For instance, an emerging and promising trend consists of modeling animals such as quadruped or birds. This is challenging due to the natural uncooperative behavior of the animals during the acquisition.

Finally, the advances on 3D registration are very important for several emerging applications. For instance, in the context of real-time interactive tasks, there is heavy expectation from new devices for modern Augmented Reality (AR), such

¹⁶See <https://www.apple.com/iphone/> <https://azure.microsoft.com/en-us/campaigns/kinect/>.

as Hololens¹⁷ or magic leap¹⁸ to mention just a few, that already integrate several sensors to improve the device pose estimation. Also, Virtual Reality (VR) frameworks require very accurate and reliable 3D tracking systems for human parts like the head and the hands to reduce the typical disadvantages of immersive devices such as sickness, or to improve the gestural interaction. Another promising application regards self-driving cars where many of the open issues mentioned above like 3D data integration, 3D mapping, and object localization are crucial. In particular, it is very important to address methods for the matching between offline data with real-time information. Registration problems are very critical also in robotics applications for effective human–robot interaction where tracking and localization should be integrated with recognition systems. Also, in this case, real-time performance is very important.

8.10 Conclusion

Registration of 3D data is a well-studied problem but still new issues need to be solved. The ICP algorithm is the current standard method since it works well in general and it is easy to implement. Although the basic version is quite limited, several extensions and strong variants have been introduced that allow it to cope with many scenarios. For instance, the techniques described in Sects. 8.2.3 and 8.4 are sufficient to obtain an automatic full model reconstruction of a single object observed from a few dozen of viewpoints. However, in more challenging situations like in the presence of cluttered or deformable objects, the problem becomes more difficult. The point matching strategy needs to be improved as well as the transformation function needs to be properly designed. Therefore, more advanced techniques need to be employed like those described in Sect. 8.3. In order to give some examples of registration algorithms, four case studies were reported. Case study 1 shows in practice how a robust outliers rejection strategy can improve the accuracy of registration and estimate the overlapping area. Case study 2 exploits general Levenberg–Marquardt optimization to improve the basic ICP algorithm. In particular, the advantage of using the distance transform is clearly demonstrated. Case study 3 addresses a more challenging problem, namely, deformable registration from real-time acquisition. Also, in this case, the Levenberg–Marquardt approach enables the modeling of the expected behavior of surface deformations. In particular, effective data and penalty terms can be encoded easily in the general error function. Finally, case study 4 shows the benefits of deformable ICP for 3D registration in laparoscopy.

¹⁷<https://www.microsoft.com/en-us/hololens>.

¹⁸<https://www.magicleap.com/>.

New challenging scenarios can be addressed as described in Sect. 8.9 by exploiting recent machine learning and computer vision techniques already successfully employed for the 2D domain as well as new advances inspired from recent computer animation techniques.

8.11 Further Reading

In order to get a more comprehensive overview of 3D registration methods, the reader can refer to milestone surveys [77, 96, 126, 128]. In [126], Ruzinkiewicz et al. have analyzed some variants of ICP techniques, focusing on methods and suggestions to improve the computation speed. An extensive review of registration methods based on the definition of surface shape descriptors can be found in [96]. In [128], Salvi et al. proposed an extensive experimental comparison among different 3D pairwise registration methods. They evaluated the accuracy of the results for both coarse and fine registrations. More recently, Kaick et al. [77] proposed a survey on shape correspondence estimation by extensively reporting and discussing interesting methods for deforming scenarios. In [165], the most recent and promising registration methods for 3D reconstruction are exhaustively reported. In particular, this survey shows how algorithms are properly designed to best exploit the benefits of using RGBD data. In [69], different methods for 6D object pose estimation from dynamic range sensors are extensively evaluated on several publicly available benchmarks. The performance of several approaches such as point-pair-based features or learning-based methods are discussed and interesting open problems are raised. In [83], the authors reported several methods for data-driven modeling and synthesis of new scenes. To this aim advanced machine learning techniques are evaluated for the integration of real examples encoded by point clouds or meshes to the process of automatic generation of new plausible objects.

The reader interested in getting in-depth details on the theoretical evaluation of registration convergence should refer to work of Pottmann et al. [58, 115]. Convergence is discussed also by Ezra et al. [52] who provided lower and upper bounds on the number of ICP iterations.

Finally, to practice the registration, the reader can evaluate the following public available tools:

- Meshlab¹⁹
- Point Cloud Library (PLC)²⁰
- Functional Automatic Registration for 3D Human Bodies (FARM).²¹

¹⁹<http://www.meshlab.net/>.

²⁰<http://pointclouds.org/>.

²¹<http://profs.scienze.univr.it/~marin/farm/>.

8.12 Questions

- Q.1 Give four examples of problem where 3D shape registration is an essential component. In each case explain why registration is required for their automated solution.
- Q.2 Briefly outline the steps of the classical Iterative Closest Points (ICP) algorithm.
- Q.3 What is usually the most computationally intensive step in a typical ICP application and what steps can be taken to reduce this?
- Q.4 What is the common failure mode of ICP and what steps can be taken to attempt to avoid this?
- Q.5 What steps can be taken to improve the final accuracy of an ICP-based registration?
- Q.6 Explain why registration in clutter is challenging and describe one solution that has been proposed.
- Q.7 Explain why registration of deformable objects is challenging and describe one solution that has been proposed.
- Q.8 Explain the effect of outliers in registration and describe one strategy that has been proposed for their detection.
- Q.9 What advantages does LM-ICP have over classical ICP?
- Q.10 Describe how DLM-ICP can be employed for computer-aided laparoscopy.

8.13 Exercises

1. Given two partial views very close to each other and an implementation of ICP,²² try to register the views by gradually moving away from the data-view from the model-view until ICP diverges. Apply the perturbation to both the translational and rotational components. Repeat the exercise, decreasing the overlap area by removing points in the model-view.
2. Implement a pairwise pre-alignment technique based on PCA. Try to check the effectiveness of the pre-alignment by varying the shape of the two views.
3. Implement an outlier rejection technique to robustify ICP registration. Compare the robustness (i) fixed threshold, (ii) threshold estimated as 2.5σ of the residuals' distribution from their mean, and (iii) threshold estimated with the X84 technique.
4. Compute the Jacobian matrix of LM-ICP by encoding rotation with quaternions.²³
5. Modify LM-ICP in order to work with multiple views, given a sequence of 10 views which surround an object such that N_{10} is highly overlapping N_1 . The global reference system is fixed on the first view. Estimate the global registration by including pairwise registration between subsequent views and by view N_{10} to view N_1 . *Suggestion:* the number of unknowns is $9p$, where p is the dimension of the transformation vector (i.e., $p = 7$ for quaternions). The number of rows of

²²A Matlab implementation can be found here <http://www.csse.uwa.edu.au/ajmal/code.html>.

²³A Matlab implementation is available here: <http://www.robots.ox.ac.uk/~awf/lmicp>.

the Jacobian matrix is given by all residual vectors of each pairwise registration. Here, the key aspect is that view N_{10} should be simultaneously aligned pairwise with both view N_9 and view N_1 .

References

1. Adagolodjo, Y., Trivisonne, R., Haouchine, N., Cotin, S., Courtecuisse, H.: Silhouette-based pose estimation for deformable organs application to surgical augmented reality. In: International Conference on Intelligent Robots and Systems (2017)
2. Aiger, D., Mitra, N.J., Cohen-Or, D.: 4-points congruent sets for robust surface registration. *ACM Trans. Graph.* **27**(3) (2008)
3. Albarelli, A., Torsello, A., Rodola, E.: A game-theoretic approach to fine surface registration without initial motion estimation. In: International Conference on Computer Vision and Pattern Recognition (2010)
4. Anguelov, D., Srinivasan, P., Pang, H.C., Koller D., Thrun, S., Davis, J.: The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In: Neural Information Processing Systems Conference (2004)
5. Arun, K.S., Huang, T., Blostein, S.: Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **9**, 698–700 (1987)
6. Avery, B., Sandor, C., Thomas, B.H.: Improving spatial perception for augmented reality x-ray vision. In: IEEE Conference on Virtual Reality (2009)
7. Bajura, M., Fuchs, H., Ohbuchi, R.: Merging virtual objects with the real world: Seeing ultrasound imagery within the patient. In: International Conference on Computer Graphics and Interactive Techniques (1992)
8. Bariya, P., Nishino, K.: Scale-hierarchical 3d object recognition in cluttered scenes. In: International Conference on Computer Vision and Pattern Recognition (2010)
9. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(4), 509–522 (2002)
10. Bernardini, F., Rushmeier, H.: The 3D model acquisition pipeline. *Comput. Graph. Forum* **21**(2), 149–172 (2002)
11. Besl, P., McKay, H.: A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(2), 239–256 (1992)
12. Bichlmeier, C., Sielhorst, T., Heining, S., Navab, N.: Improving depth perception in medical AR: a virtual vision panel to the inside of the patient. In: *Bildverarbeitung für die Medizin* (2007)
13. Blais, G., Levine, M.: Registering multiview range data to create 3d computer objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **17**(8) (1995)
14. Bogo, F., Black, M.J., Loper, M., Romero, J.: Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In: International Conference on Computer Vision (2015)
15. Bogo, F., Romero, J., Pons-Moll, G., Black, M.J.: Dynamic FAUST: registering human bodies in motion. In: International Conference on Computer Vision and Pattern Recognition (2017)
16. Boscaini, D., Masci, J., Melzi, S., Bronstein, M., Castellani, U., Vandergheynst, P.: Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. *Comput. Graph. Forum* **34**(5) (2015)
17. Bouaziz, S., Tagliasacchi, A., Pauly, M.: Sparse iterative closest points. *Comput. Graph. Forum* **32**(5) (2013)
18. Bowyer, K.W., Chang, K., Flynn, P.: A survey of approaches and challenges in 3d and multi-modal 3d + 2d face recognition. *Comput. Vis. Image Underst.* **101**(1) (2006)

19. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3D shape from image streams. In: International Conference on Computer Vision and Pattern Recognition (2000)
20. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Three-dimensional face recognition. *Int. J. Comput. Vis.* **64**(1), 5–30 (2005)
21. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: *Numerical Geometry of Non-rigid Shapes*. Springer, Berlin (2008)
22. Brown, B., Rusinkiewicz, S.: Non-rigid range-scan alignment using thin-plate splines. In: Symposium on 3D Data Processing, Visualization, and Transmission (2004)
23. Brown, B., Rusinkiewicz, S.: Global non-rigid alignment of 3-D scans. *ACM Trans. Graph. (Proc. SIGGRAPH)* **26**(3) (2007)
24. Brown, M., Lowe, D.: Recognising panoramas. In: ICCV (2003)
25. Brusco, N., Andreetto, M., Giorgi, A., Cortelazzo, G.: 3d registration by textured spin-images. In: 3DIM '05: Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling, pp. 262–269 (2005)
26. Campbell, R., Flynn, P.: A survey of free-form object representation and recognition techniques. *Comput. Vis. Image Underst.* **81**(2), 166–210 (2001)
27. Castellani, U., Cristani, M., Fantoni, S., Murino, V.: Sparse points matching by combining 3D mesh saliency with statistical descriptors. In: Computer Graphics Forum, vol. 27, pp. 643–652. Blackwell Publishing (2008)
28. Castellani, U., Cristani, M., Murino, V.: Statistical 3d shape analysis by local generative descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(12) (2011)
29. Castellani, U., Fusiello, A., Murino, V.: Registration of multiple acoustic range views for underwater scene reconstruction. *Comput. Vis. Image Underst.* **87**(3), 78–89 (2002)
30. Castellani, U., Fusiello, A., Murino, V., Papaleo, L., Puppo, E., Pittore, M.: A complete system for on-line modelling of acoustic images. *Image Commun. J.* **20**(9–10), 832–852 (2005)
31. Castellani, U., Gay-Bellile, V., Bartoli, A.: Robust deformation capture from temporal range data for surface rendering. *Comput. Animat. Virtual Worlds* **19**(5), 591–603 (2008)
32. Chang, M., Leymarie, F., Kimia, B.: 3d shape registration using regularized medial scaffolds. In: International Symposium on 3D Data Processing, Visualization and Transmission (2004)
33. Chang, W., Zwicker, M.: Automatic registration for articulated shapes. *Comput. Graph. Forum (Proc. SGP 2008)* **27**(5), 1459–1468 (2008)
34. Chang, W., Zwicker, M.: Range scan registration using reduced deformable models. *Comput. Graph. Forum* **28**(2), 447–456 (2009)
35. Chauvet, P., Collins, T., Debize, C., Novais-Gameiro, L., Pereira, B., Bartoli, A., Canis, M., Bourdel, N.: Augmented reality in a tumor resection model. *Surg. Endosc.* **32**(3), 1192–1201 (2018)
36. Chen, Y., Medioni, G.: Object modelling by registration of multiple range images. *Image Vis. Comput.* **10**(3), 145–155 (1992)
37. Choi, S., Zhou, Q.Y., Koltun, V.: Robust reconstruction of indoor scenes. In: International Conference on Computer Vision and Pattern Recognition (2015)
38. Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. *Comput. Vis. Image Underst.* **89**(2–3), 114–141 (2003)
39. Collins, T., Pizarro, D., Bartoli, A., Bourdel, N., Canis, M.: Computer-aided laparoscopic myomectomy by augmenting the uterus with pre-operative MRI data. In: International Symposium on Mixed and Augmented Reality (2014)
40. Collins, T., Pizarro, D., Bartoli, A., Canis, M., Bourdel, N.: Realtime wide-baseline registration of the uterus in monocular laparoscopic videos. In: International Workshop on Medical Imaging and Augmented Reality at MICCAI (2013)
41. Corey, G., Matei, C., Jaime, P., Hao, D., K., A.P.: Data-driven grasping with partial sensor data. In: IROS'09: Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems, pp. 1278–1283 (2009)
42. Cruska, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV Workshop on Statistical Learning in Computer Vision, pp. 1–22 (2004)

43. Cunnington, S., Stoddart, A.: N-view point set registration: a comparison. In: British Machine Vision Conference (1999)
44. Deng, H., Birdal, T., Ilic, S.: PPFNet: Global context aware local features for robust 3d point matching. In: International Conference on Computer Vision and Pattern Recognition (2018)
45. Dewaele, G., Devernay, F., Horaud, R.: Hand motion from 3d point trajectories and a smooth surface model. In: European Conference on Computer Vision (2004)
46. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: Efficient and robust 3d object recognition. In: International Conference on Computer Vision and Pattern Recognition (2010)
47. Eckart, B., Kim, K., Kautz, J.: HGMR: Hierarchical gaussian mixtures for adaptive 3d registration. In: European Conference on Computer Vision (2018)
48. Eckart, B., Kim, K., Troccoli, A., Kelly, A., Kautz, J.: MLMD: maximum likelihood mixture decoupling for fast and accurate point cloud registration. In: Proceedings of the International Conference on 3D Vision (2015)
49. Eggert, D., Lorusso, A., Fisher, R.: Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Mach. Vis. Appl.* **9**, 272–290 (1997)
50. Elbaz, G., Avraham, T., Fischer, A.: 3d point cloud registration for localization using a deep neural network auto-encoder. In: International Conference on Computer Vision and Pattern Recognition (2017)
51. Endres, F., Hess, J., Sturm, J., Cremers, D., Burgard, W.: 3D mapping with an RGB-D camera. *IEEE Trans. Robot.* **30**(1), (2014)
52. Ezra, E., Sharir, M., Efrat, A.: On the performance of the ICP algorithm. *Comput. Geom.* **41**(1–2), 77–93 (2008)
53. Fantoni, S., Castellani, U., Fusiello, A.: Accurate and automatic alignment of range surfaces. In: Proceedings of the 2nd Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT) (2012)
54. Fitzgibbon, A.: Robust registration of 2D and 3D point sets. *Image Vis. Comput.* **21**(13–14), 1145–1153 (2003)
55. Fuchs, H., Livingston, M., Raskar, R., Colucci, D., Keller, K., State, A., Crawford, J., Rademacher, P., Drake, S., Meyer, A.: Augmented reality visualization for laparoscopic surgery. In: *Medical Image Computing and Computer-Assisted Intervention* (1998)
56. Funkhouser, T., Kazhdan, M., Min, P., Shilane, P.: Shape-based retrieval and analysis of 3d models. *Commun. ACM* **48**(6), 58–64 (2005)
57. Garcia-Grasa, O., Bernal, E., Casado, S., Gil, I., Montiel, J.M.M.: Visual SLAM for hand-held monocular endoscope. *IEEE Trans. Med. Imaging* **33**(1), 135–146 (2014)
58. Gelfand, N., Mitra, N.J., Guibas, L.J., Pottmann, H.: Robust global registration. In: Desbrun, M., Pottmann, H. (eds.), *Eurographics Association*, pp. 197–206 (2005). ISBN 3-905673-24-X
59. Godin, G., Laurendeau, D., Bergevin, R.: A method for the registration of attributed range images. In: *3-D Digital Imaging and Modeling (3DIM)*, pp. 179–186 (2001)
60. Golovinskiy, A., Kim, V., Funkhouser, T.: Shape-based recognition of 3d point clouds in urban environments. In: International Conference on Computer Vision (2009)
61. Granger, S., Pennec, S.: Multi-scale EM-ICP: a fast and robust approach for surface registration. In: European Conference on Computer Vision (2002)
62. Gu, X., Gortler, S.J., Hoppe, H.: Geometry images. *ACM Trans. Graph.* **21**(3), 355–361 (2002)
63. Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J.: 3d object recognition in cluttered scenes with local surface features: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(11) (2014)
64. Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J., Kwok, N.M.: A comprehensive performance evaluation of 3d local feature descriptors. *Int. J. Comput. Vis.* **116**(1) (2016)
65. Guo, Y., Sohel, F., Bennamoun, M., Wan, J., Lu, M.: An accurate and robust range image registration algorithm for 3d object modeling. *IEEE Trans. Multimed.* **16**(5) (2014)
66. Hampel, F., Rousseeuw, P., Ronchetti, E., Stahel, W.: *Robust Statistics: The Approach Based on Influence Functions*. Wiley (1986)

67. Handa, A., Whelan, T., McDonald, J., Davison, A.: A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In: *IEEE International Conference on Robotics and Automation, ICRA* (2014)
68. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. *Science* **313**(5786) (2006)
69. Hodaň, T., Michel, F., Brachmann, E., Kehl, W., Buch, A.G., Kraft, D., Drost, B., Vidal, J., Ihrke, S., Zabulis, X., Sahin, C., Manhardt, F., Tombari, F., Kim, T.K., Matas, J., Rother, C.: BOP: Benchmark for 6d object pose estimation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) *European Conference on Computer Vision* (2018)
70. Horaud, R., Forbes, F., Yguel, M., Dewaele, G., Zhang, J.: Rigid and articulated point registration with expectation conditional maximization. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**, 587–602 (2011)
71. Huang, Q., Adams, B., Wicke, M., Guibas, L.: Non-rigid registration under isometric deformations. *Comput. Graph. Forum* **27**(5), 1449–1457 (2008)
72. Huber, D., Hebert, M.: Fully automatic registration of multiple 3D data sets. *Image Vis. Comput.* **21**(7), 637–650 (2003)
73. Innmann, M., Zollhöfer, M., Nießner, M., Theobalt, C., Stamminger, M.: Volumedeform: Real-time volumetric non-rigid reconstruction. In: *European Conference on Computer Vision* (2016)
74. Jhonson, A., Kang, S.: Registration and integration of textured 3d data. *Image Vis. Comput.* **19**(2), 135–147 (1999)
75. Jian, B., Vemuri, B.C.: A robust algorithm for point set registration using mixture of gaussians. In: *International Conference on Computer Vision and Pattern Recognition* (2005)
76. Johnson, A., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(5), 433–449 (1999)
77. van Kaick, O., Zhang, H., Hamarneh, G., Cohen-Or, D.: A survey on shape correspondence. In: *EuroGraphics: State-of-the-Art Report* (2010)
78. Khoualed, S., Castellani, U., Bartoli, A.: Semantic shape context for the registration of multiple partial 3-D views. In: *British Machine Vision Conference* (2009)
79. Koo, B., Özgür, E., Roy, B.L., Buc, E., Bartoli, A.: Deformable registration of a preoperative 3D liver volume to a laparoscopy image using contour and shading cues. In: *Medical Image Computing and Computer-Assisted Intervention* (2017)
80. Krsek, P., Pajdla, T., Hlaváč, V.: Differential invariants as the base of triangulated surface registration. *Comput. Vis. Image Underst.* **87**(1–3), 27–38 (2002)
81. Krull, A., Brachmann, E., Michel, F., Yang, M.Y., Gumhold, S., Rother, C.: Learning analysis-by-synthesis for 6d pose estimation in RGB-D images. In: *International Conference on Computer Vision* (2015)
82. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(436) (2015)
83. Lescoat, T., Ovsjanikov, M., Memari, P., Thiery, J.M., Boubekur, T.: A survey on data-driven dictionary-based methods for 3d modeling. *Comput. Graph. Forum* **37**(2) (2018)
84. Lévy, B.: Laplace-Beltrami eigenfunctions towards an algorithm that understands geometry. In: *Proceedings of SMI*, pp. 13–25. IEEE, Washington, DC (2006)
85. Li, H., Sumner, R.W., Pauly, M.: Global correspondence optimization for non-rigid registration of depth scans. *Comput. Graph. Forum (Proc. SGP'08)* **27**(5) (2008)
86. Litany, O., Remez, T., Rodolá, E., Bronstein, A.M., Bronstein, M.M.: Functional maps: structured prediction for dense shape correspondence. In: *International Conference on Computer Vision* (2017)
87. Litany, O., Rodolá, E., Bronstein, A.M., Bronstein, M.M., Cremers, D.: Non-Rigid Puzzles. *Comput. Graph. Forum* **35**(5), 135–143 (2016)
88. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. *ACM Trans. Graph.* **34**(6) (2015)
89. Maintz, J., Viergever, M.A.: A survey of medical image registration. *Med. Image Anal.* **2**(1), 1–36 (1998)

90. Makadia, A., Patterson, A., Daniilidis, K.: Fully Automatic Registration of 3D Point Clouds. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Volume 1, pp. 1297–1304. IEEE Computer Society Washington, DC, USA (2006)
91. Marin, R., Melzi, S., Rodolà, E., Castellani, U.: Farm: Functional automatic registration method for 3d human bodies. (2018). [arXiv:1807.10517](https://arxiv.org/abs/1807.10517)
92. Marzotto, R., Fusiello, A., Murino, V.: High resolution video mosaicing with global alignment. In: ICCV (2004)
93. Mellado, N., Aiger, D., Mitra, N.J.: Super 4PCS fast global pointcloud registration via smart indexing. *Comput. Graph. Forum* **33**(5) (2014)
94. Melzi, S., Rodolà, E., Castellani, U., Bronstein, M.M.: Localized manifold harmonics for spectral shape analysis. *Comput. Graph. Forum* **37**(6), 20–34 (2018)
95. Mian, A.S., Bennamoun, M., Owens, R.: Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(10), 1584–1601 (2006)
96. Mian, A.S., Bennamoun, M., Owens, R.A.: Automatic correspondence for 3d modeling: an extensive review. *Int. J. Shape Model.* **11**(2), 253–291 (2005)
97. Mitra, N.J., Flory, S., Ovsjanikov, M., Gelfand, N., Guibas, L., Pottmann, H.: Dynamic geometry registration. In: Symposium on Geometry Processing, pp. 173–182 (2007)
98. Munoz, D., Vandapel, N., Hebert, M.: Directional associative markov network for 3-d point cloud classification. In: International Symposium on 3-D Data Processing, Visualization, and Transmission (3DPVT) (2008)
99. Murino, V., Ronchetti, L., Castellani, U., Fusiello, A.: Reconstruction of complex environments by robust pre-aligned ICP. In: 3DIM (2001)
100. Myronenko, A., Song, X., Carreira-Perpinan, M.: Non-rigid point set registration: Coherent point drift. In: Neural Information Processing Systems Conference (2006)
101. Newcombe, R., Davison, A., Izadi, S., Kohli, P., O., H., Shotton, J., Molyneaux, D., Hodges, S., Kim, D., Fitzgibbon, A.: Kinectfusion: real-time dense surface mapping and tracking. In: IEEE ISMA (2011)
102. Newcombe, R.A., Fox, D., Seitz, S.M.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In: International Conference on Computer Vision and Pattern Recognition (2015)
103. Nogneng, D., Melzi, S., Rodolà, E., Castellani, U., Bronstein, M., Ovsjanikov, M.: Improved functional mappings via product preservation. *Comput. Graph. Forum* **37**(2), 179–190 (2018)
104. Novatnack, J., Nishino, K.: Scale-Dependent/Invariant Local 3D Shape Descriptors for Fully Automatic Registration of Multiple Sets of Range Images. In: Proceedings of the 10th European Conference on Computer Vision: Part III, pp. 440–453. Springer, Berlin, Heidelberg (2008)
105. Nuchter, A., Lingemann, K., Hertzberg, J.: Cached k-d tree search for ICP algorithms. In: 3DIM '07: Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling, pp. 419–426 (2007)
106. Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L.: Functional maps: a flexible representation of maps between shapes. *ACM Trans. Graph. (TOG)* **31**(4) (2012)
107. Özgür, E., Lafont, A., Bartoli, A.: Visualizing in-organ tumors in augmented monocular laparoscopy. In: International Symposium on Mixed and Augmented Reality (2017)
108. Parashar, S., Pizarro, D., Bartoli, A.: Isometric non-rigid shape-from-motion with Riemannian geometry solved in linear time. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(10), 2442–2454 (2018)
109. Park, K., Germann, M., Breitenstein, M.D., Pfister, H.: Fast and automatic object pose estimation for range images on the GPU. *Mach. Vis. Appl.* **21**(5), 749–766 (2009)
110. Park, S., Subbarao, M.: An accurate and fast point-to-plane registration technique. *Pattern Recognit. Lett.* **24**(16), 2967–2976 (2003)
111. Patterson IV, A., Mordohai, P., Daniilidis, K.: Object detection from large-scale 3d datasets using bottom-up and top-down descriptors. In: Proceedings of the European Conference on Computer Vision (2008)

112. Pears, N.E., Heseltine, T., Romero, M.: From 3d point clouds to pose normalised depth maps. *Int. J. Comput. Vis.* **89**(2), 152–176 (2010)
113. Phillips, J., Liu, R., Tomasi, C.: Outlier robust ICP for minimizing fractional RMSD. In: 3-D Digital Imaging and Modeling (3DIM), pp. 427–434 (2007)
114. Pissanetzky, S.: *Sparse Matrix Technology*. Academic, New York (1984)
115. Pottmann, H., Huang, Q., Yang, Y., Hu, S.: Geometry and convergence analysis of algorithms for registration of 3D shapes. *Int. J. Comput. Vis.* **67**(3), 277–296 (2006)
116. Prasad, M., Zisserman, A., Fitzgibbon, A.W.: Single view reconstruction of curved surfaces. In: *International Conference on Computer Vision and Pattern Recognition* (2006)
117. Pulli, K.: Multiview registration for large data sets. In: 3DIM: Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling, pp. 160–168 (1999)
118. Pulli, K., Piironen, S., Duchamp, T., Stuetzle, W.: Projective surface matching of colored 3d scans. In: 3-D Digital Imaging and Modeling (3DIM), pp. 531–538 (2005)
119. Rangarajan, A., Chui, H., Duncan, J.: Rigid point feature registration using mutual information. *Med. Image Anal.* **3**, 425–440 (1999)
120. Rangarajan, A., Chui, H., Mjolsness, E., Pappu, S., Davachi, L., Goldman-Rakic, P., Duncan, J.: A robust point-matching algorithm for autoradiograph alignment. *Med. Image Anal.* **1**, 379–398 (1997)
121. Rethage, D., Wald, J., Sturm, J., Navab, N., Tombari, F.: Fully-convolutional point networks for large-scale point clouds. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) *European Conference on Computer Vision* (2018)
122. Rodolà, E., Cosmo, L., Bronstein, M.M., Torsello, A., Cremers, D.: Partial functional correspondence. *Comput. Graph. Forum* **36**(1), 222–236 (2017)
123. Ruitter, H.D., Benhabib, B.: *On-line Modeling for Real-Time, Model-Based, 3D Pose Tracking*. Springer, Berlin (2007)
124. Rusinkiewicz, S., Brown, B., Kazhdan, M.: 3d scan matching and registration. *ICCV Short Course* (2005)
125. Rusinkiewicz, S., Hall-Holt, O., Levoy, M.: Real-time 3-D model acquisition. *ACM Trans. Graph. (Proc. SIGGRAPH)* **21**(3), 438–446 (2002)
126. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: 2001. Proceedings Third International Conference on 3-D Digital Imaging and Modeling, pp. 145–152 (2001)
127. Russell, C., Yu, R., Agapito, L.: Video pop-up: Monocular 3D reconstruction of dynamic scenes. In: *European Conference on Computer Vision* (2014)
128. Salvi, J., Matabosch, C., Fofi, D., Forest, J.: A review of recent range image registration methods with accuracy evaluation. *Image Vis. Comput.* **25**(5), 578–596 (2007)
129. Salzmann, M., Ilic, S., Fua, P.: Physically valid shape parameterization for monocular 3-D deformable surface tracking. In: *British Machine Vision Conference* (2005)
130. Sara, R.: Finding the largest unambiguous component of stereo matching. In: Proceedings of European Conference on Computer Vision (ECCV), pp. 900–914 (2002)
131. Sara, R., Okatani, I., Sugimoto, A.: Globally convergent range image registration by graph kernel algorithm. In: 3-D Digital Imaging and Modeling (3DIM) (2005)
132. Sawhney, H.S., Hsu, S., Kumar, R.: Robust video mosaicing through topology inference and local to global alignment. In: *ECCV* (1998)
133. Scheenstra, A., Ruifrok, A., Veltkamp, R.C.: A survey of 3d face recognition methods. In: *Audio- and Video-Based Biometric Person Authentication*, pp. 891–899 (2005)
134. Shams, R., Sadeghi, P., Kennedy, R.A., Hartley, R.I.: A survey of high performance medical image registration on multi-core, GPU and distributed architectures. *IEEE Signal Process. Mag.* **27**(2), 50–60 (2010)
135. Sharp, G., Sang, L., Wehe, D.: ICP registration using invariant features. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(1), 90–102 (2002)
136. Simon, D.A.: Fast and accurate shape-based registration. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA (1996)

137. Simonovsky, M., Gutiérrez-Becker, B., Mateus, D., Navab, N., Komodakis, N.: A deep metric for multimodal registration. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016* (2016)
138. Sorkine, O., Alexa, M.: As-rigid-as-possible surface modeling. *Symp. Geom. Process.* **4**, 109–116 (2007)
139. Steinke, F., Scholkopf, B., Blanz, V.: Learning dense 3d correspondence. In: *Annual Conference on Neural Information Processing Systems (NIPS 2006)* (2007)
140. Taati, B., Bondy, M., Jasiobedzki, P., Greenspan, M.: Automatic registration for model building using variable dimensional local shape descriptors. In: *International Conference on 3-D Digital Imaging and Modeling* (2007)
141. Tabia, H., Guo, Y., Laga, H., Fisher, R.B., Bennamoun, M.: *3D Shape Analysis: Fundamentals, Theory, and Applications*. Wiley, New York (2019)
142. Tam, G.K., Cheng, Z.Q., Lai, Y.K., Langbein, F.C., Liu, Y., Marshall, D., Martin, R.R., Sun, X.F., Rosin, P.L.: Registration of 3d point clouds and meshes: a survey from rigid to nonrigid. *IEEE Trans. Vis. Comput. Graph.* **19**(7), 1199–1217 (2012)
143. Tangelder, J., Veltkamp, R.: A survey of content based 3d shape retrieval methods. *Multimed. Tools Appl.* **39**(3), 441–471 (2008)
144. Toldo, R., Beinat, A., Crosilla, F.: Global registration of multiple point clouds embedding the generalized procrustes analysis into an ICP framework. In: *Symposium on 3D Data Processing, Visualization, and Transmission* (2010)
145. Tombari, F., Salti, S., Stefano, L.D.: Performance evaluation of 3d keypoint detectors. *Int. J. Comput. Vis.* **102**(1–3) (2013)
146. Tombari, F., Salti, S., Stefano, L.D.: SHOT: Unique signatures of histograms for surface and texture description. *Int. J. Comput. Vis.* **125** (2014)
147. Trucco, M., Verri, A.: *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, Upper Saddle River (1998)
148. Tsing, Y., Kanade, T.: A correlation-based approach to robust point set registration. In: *European Conference on Computer Vision*, pp. 558–569 (2004)
149. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(4), 376–380 (1991)
150. Vinesh, R., Kiran, F.: *Reverse Engineering, an Industrial Perspective*. Springer, Berlin (2008)
151. Wang, F., Vemuri, B.C., Rangarajan, A.: Groupwise point pattern registration using a novel CDF-based Jensen-Shannon Divergence. In: *International Conference on Computer Vision and Pattern Recognition* (2006)
152. Wang, R., Geng, Z., Zang, Z., Pei, R.: Visualization techniques for augmented reality in endoscopic surgery. In: *Medical Imaging and Augmented Reality Workshop at MICCAI* (2016)
153. Watt, A.: *3D Computer Graphics*. Addison-Wesley, Boston (2000)
154. Weik, S.: Registration of 3-d partial surface models using luminance and depth information. In: *3-D Digital Imaging and Modeling (3DIM)*, pp. 93–100 (1997)
155. Woodford, O.J., Pham, M.T., Maki, A., Perbet, F., Stenger, B.: Demisting the hough transform for 3d shape recognition and registration. *Int. J. Comput. Vis.* **106**(3), (2014)
156. Wyngaerd, J.V., Gool, L.V.: Automatic crude patch registration: Toward automatic 3d model building. *Comput. Vis. Image Underst.* **87**(1–3), 8–26 (2002)
157. Yang, J., Li, H., Campbell, D., Jia, Y.: Go-ICP: a globally optimal solution to 3d ICP point-set registration. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(11) (2016)
158. Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T.: 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In: *International Conference on Computer Vision and Pattern Recognition* (2017)
159. Zhang, Z.: Iterative point matching of free-form curves and surfaces. *Int. J. Comput. Vis.* **13**, 119–152
160. Zhou, L., Zhu, S., Luo, Z., Shen, T., Zhang, R., Zhen, M., Fang, T., Quan, L.: Learning and matching multi-view descriptors for registration of point clouds. In: *European Conference on Computer Vision* (2018)

161. Zhou, Q.Y., Koltun, V.: Dense scene reconstruction with points of interest. *ACM Trans. Graph.* **32**(4) (2013)
162. Zhou, Q.Y., Koltun, V.: Color map optimization for 3d reconstruction with consumer depth cameras. *ACM Trans. Graph.* **33**(4) (2014)
163. Zhou, Q.Y., Park, J., Koltun, V.: Fast global registration. In: *European Conference on Computer Vision* (2016)
164. Zinsser, T., Schmidt, H., Niermann, J.: A refined ICP algorithm for robust 3D correspondences estimation. In: *International Conference on Image Processing*, pp. 695–698 (2003)
165. Zollhöfer, M., Stotko, P., Görlitz, A., Theobalt, C., Nießner, M., Klein, R., Kolb, A.: State of the Art on 3D Reconstruction with RGB-D Cameras. *Comput. Graph. Forum* **37**(2) (2018)
166. Zuffi, S., Black, M.J.: The stitched puppet: A graphical model of 3D human shape and pose. In: *International Conference on Computer Vision and Pattern Recognition* (2015)