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. Three 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 optimization procedure to solve standard pairwise registration. The third case study focuses on the challenging problem of deformable object registration. Finally, open issues and directions for future work are discussed and conclusions are drawn.

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:

University of Verona
Verona, Italy – e-mail: Umberto.Castellani@univr.it

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

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 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 figure 1). In this context registration is first applied between pairs of views [6, 75]. The whole model is then reconstructed using multiple view registration refinement [75, 40]. Typically, model reconstruction is employed in *cultural heritage* [5] to obtain 3D models of archaeological findings. It has also been applied in applications such as *reverse engineering* and *rapid prototyping* [91] and for vision in hostile environments [16, 17].
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 [61, 24] and model-based object tracking [72]. Model fitting is typically used with rigid objects but has recently been extended to deformable objects [18].
3. **Object recognition**. The goal in object recognition is to find, amongst 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* [89]. Several works have been done for 3D face recognition [9, 8, 80] and for 3D object retrieval [31, 87]. Registration becomes more challenging in a cluttered environment [53, 44, 3].
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 common to register MRI and CT scans or MRI and PET scans [51, 81]. 3D medical image registration is discussed further in Chapter 11.

This chapter gives a general formulation for 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 [75, 73, 54, 45, 76]. It is worth mentioning that most of the existing computational solutions are based on the seminal Iterative Closest Point (ICP)[6] algorithm that we will describe shortly.

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

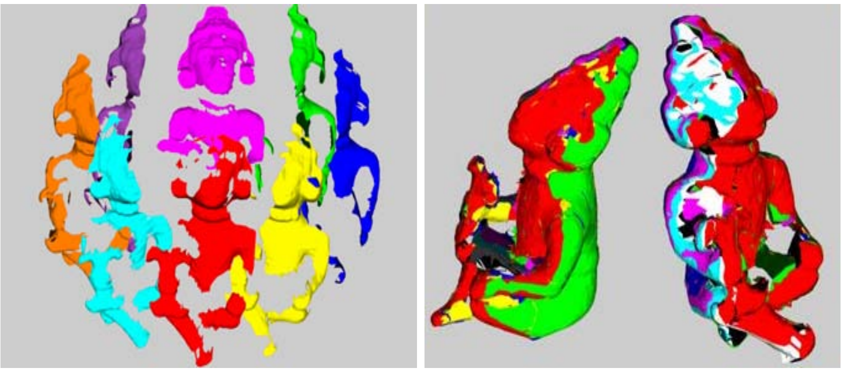


Fig. 1 Example of model reconstruction. Partial 3D views of the object of interest are acquired (left). After registration all the 3D views are transformed to the common reference system and merged (right).

by proposing three case studies. We give an overview of open challenges with future directions and conclusion. Further suggestions, additional reading and exercises are finally proposed.

2 Registration of Two Views

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

2.1 Problem Statement

Given a pair of views D and 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, D and M are either simple point clouds or triangulated meshes [14]. The moving view D is called *data-view*, while the fixed view is called *model-view*. The registration problem is solved by estimating the parameters $\mathbf{a}^*$ of the transformation T that satisfy:

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} E(T(\mathbf{a}, D), M), \tag{1}$$

where E is called the *error function* and measures the registration error. Figure 2 illustrates the two-view registration process. The data-view and the model-view show different portions of *Bunny*. The transformation function $T(\mathbf{a},D)$ is applied and the registered views are shown.

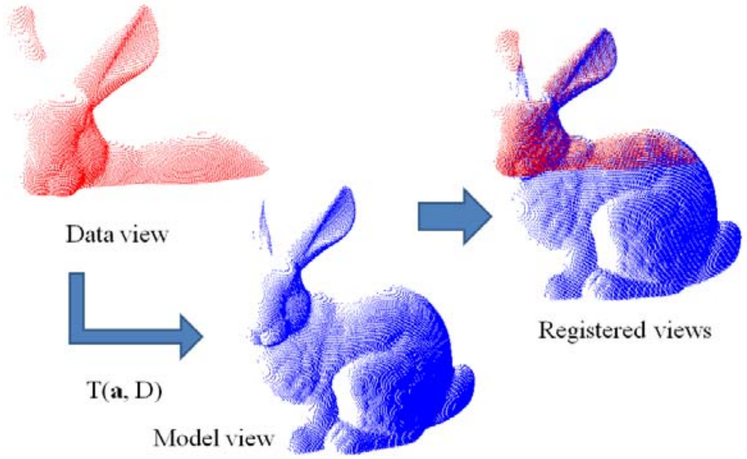


Fig. 2 Pairwise registration. The data-view and the model-view (left) are registered. The transformation function $T(\mathbf{a},D)$ allows one to move the data-view to the model-view coordinate frame (right).

Most of the registration methods are based on the paradigm defined directly above and differ o 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 D and 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 [6] or the *point-to-plane* distance [22].
- **The optimisation method**. This is the method or algorithm used to find the minimizer $\mathbf{a}$ in problem (1). The gold standard is the ICP algorithm [6] which was specifically designed for the problem at hand. General purpose optimisation methods such as Levenberg-Marquardt [30] have also been used for this problem.

2.2 Iterative Closest Point (ICP)

In the classical ICP algorithm [6] the overall aim is to estimate a rigid transformation with parameters $\mathbf{a}^* = (\mathbf{R},\mathbf{t})$. Both views are treated as point clouds $D = \{\mathbf{d}_1,\dots,\mathbf{d}_{N_d}\}$ and $M = \{\mathbf{m}_1,\dots,\mathbf{m}_{N_m}\}$. The error function is chosen as:

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

where we define $E_{ICP}(\mathbf{a},D,M) = E(T(\mathbf{a},D),M)$ and where $(\mathbf{d}_i,\mathbf{m}_j)$ are corresponding points [75][1]. Fixing $\mathbf{d}_i \in D$ the corresponding point $\mathbf{m}_j \in M$ is computed such that:

$$j = \underset{j\in\{1,\dots,N_m\}}{\arg\min} \|(\mathbf{R}\mathbf{d}_i + \mathbf{t}) - \mathbf{m}_j\|^2. \tag{3}$$

More specifically, the value

$$e_i^2 = \|(\mathbf{R}\mathbf{d}_i + \mathbf{t}) - \mathbf{m}_j\|^2 \tag{4}$$

is the square of the *residual*. Figure 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, problem (2) can be solved in closed-form [75].

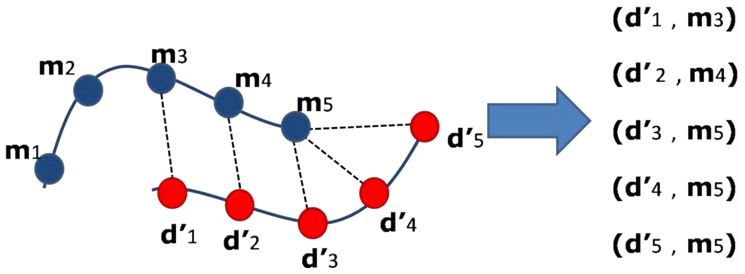


Fig. 3 Correspondence estimation in ICP. For each data point $\mathbf{d}'_i = (\mathbf{R}\mathbf{d}_i + \mathbf{t})$ the closest model point $\mathbf{m}_j$ is estimated (left). The list of corresponding points is then defined (right).

The ICP algorithm is iterative because it iteratively improves the putative correspondences. If true corresponences were known, clearly the process could operate

[1] Note that the pair $(\mathbf{d}_i,\mathbf{m}_j)$ is a putative correspondence which becomes true correspondence when convergence is reached.

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 details, the algorithm is:

1. For each data-point $\mathbf{d}_i \in D$, compute the closest point $\mathbf{m}_j \in M$ according to equation 3.
2. With the correspondences $(\mathbf{d}_i, \mathbf{m}_j)$ from step 1, compute 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 D,
4. If the change in $E_{ICP}(\mathbf{a}, D, M)$ between two successive iterations is lower than a threshold terminate, else goto step 1.

It was proven [6] that this algorithm is guaranteed to converge monotonically to a local solution of problem (2). Note that as for any local iterative method, a strategy for initializing **a** must be used. An overview of the most popular initialization strategies is given in section 2.3.1.

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 D must be a subset of the model-view 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 the speed of convergence by proposing new distance formulations for problem (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 4 illustrates the proposed taxonomy of ICP extensions so as to easily understand the organization of previous work in this field.

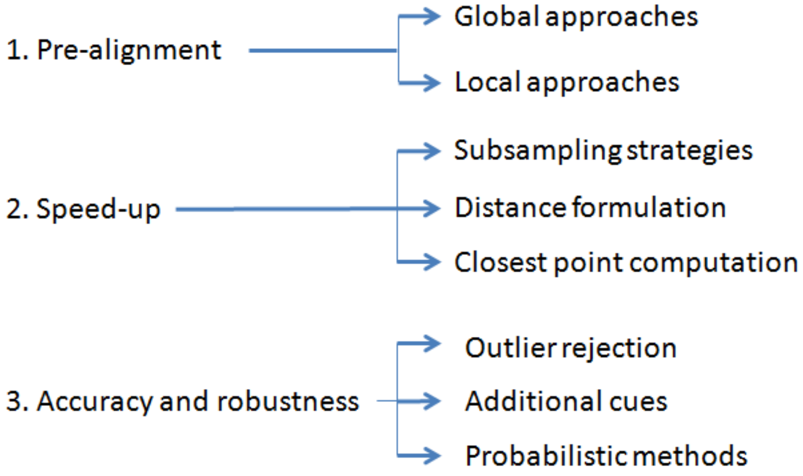


Fig. 4 A taxonomy of some ICP extensions.

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 subparts 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 PCA (Principal Component Analysis) to the 3D points can be performed. The problems with PCA as a pre-alignment method are (i) a 180 degree ambiguity in the 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 (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 degree ambiguity, unless higher order moments are used. Moments of higher orders are also useful to improve accuracy [10]. 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 [57, 19] but this is feasible for articulated objects only.

Local approaches

Local approaches define a *descriptor* (or signature) for each 3D point which encodes local shape variation in the point neighborhood [44, 15, 59, 46, 86]. Point correspondences are then obtained as the best matches in regard of the point signatures. Various methods to compute signatures were proposed. In the seminal work [44], the *Spin Images* were introduced. In a spin-image, the neighbours of some selected 3D point (e.g. a 3D interest point) 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 [47] 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 [95] bitangent curve pairs were used as landmarks on the surface. In [59] a geometric scale-space analysis of 3D models was proposed from which a scale-dependent local shape descriptor was derived. Similarly in [15] 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 [46] 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* [4] which was improved using the *Bag-of-Words* paradigm [25]. 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 [79] 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* [78].

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.

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 [75]. *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.

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 [84] structure in order to reduce the closest point complexity to $O(n \log n)$. Closest point caching [84] 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 [60] *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 [7]. The idea is to project the source data point onto the destination model-view which is encoded as a *range* image [73]. 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 application. For instance in [74] the authors proposed a real-time 3D model reconstruction system, and in [17] on-line registration is performed to build a 3D mosaic of the scene in order to improve the navigation in underwater environments. The one-shot computation can be carried out also on generic point cloud (not necessary coming from a range image) by precomputing the so called *distance transform* of the model view [30]. Figure 5 illustrates the distance transform. In practice the distance to closest model-points are precomputed 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 istances of data scan to the same model scan.

Distance formulation

Another crucial factor affecting the speed of ICP is the point-to-point or point-to-plane distance used in problem (1). Figure 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), point-to-plane distance computes the projection of the data-point onto the surface of the the model-view which is encoded in terms of piecewise planar patches (for instance a triangular mesh). In spite of an increased complexity

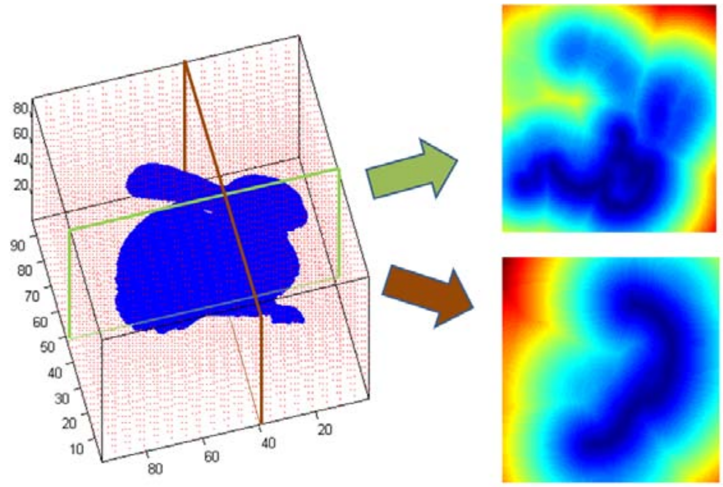


Fig. 5 Using the distance transform. The model-view is enclosed in a volumetric grid (left). For each point of the grid the closest model-point is computed. Two planes are highlighted on the XY and YZ axes respectively and the distance transform values of each grid-point are visualized for both planes (right).

of the distance formulation, the number of ICP iterations required to converge is reduced [62, 66]. Whether this results in a reduced registration time depends on the trade-off between the increased per-iteration time and the reduced number of iterations. Results regarding this aspect on example 3D scans are presented in section 4.

Recently a new “distance formulation” has been proposed [63] where the model surface is implicitly represented as the zero-isosurface of a fitted radial basis function (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 $O(n \log n)$.

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

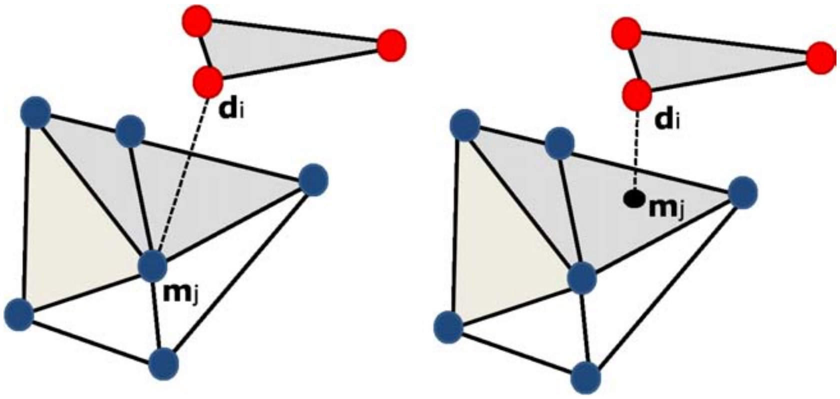


Fig. 6 Distance formulation. Point-to-point distance (left) and point-to-plane distance (right).

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 techniques threshold the residuals. The threshold can be fixed manually, or as a percentage of worst pairs (e.g., 10% [75, 68]). Other techniques perform statistics on the residual vector and set the threshold as 2.5σ or apply the so-called *X84* rule [16, 37]. More recently, statistical analysis has been introduced into the general registration problem (equation 1) by proposing a new error function named *Fractional Root Mean Squared Distance* [64].

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 [33], intensity derived from the sensor [94, 33], or color [69]. In [42] the authors proposed to use color and texture information. In [82] the so-called *ICP using invariant features* (ICPIF) was introduced where several geometric features are employed, namely *curvatures*, *moments invariants* and *Spherical Harmonics Invariants*. In [13] additional information was integrated in the point descriptors using the Spin Image with color.

Probabilistic method

In order to improve the robustness of the registration several probabilistic version of the standard ICP have been proposed [71, 70, 35]. In [71, 70] 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 [71]) and Mutual Information [70], leading to a smaller number of local minima and thus presenting the most convincing improvements. In [35] 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.

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 the 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 7 illustrates the proposed taxonomy for advanced registration techniques.

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. 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 $N_1,...,N_p$ 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

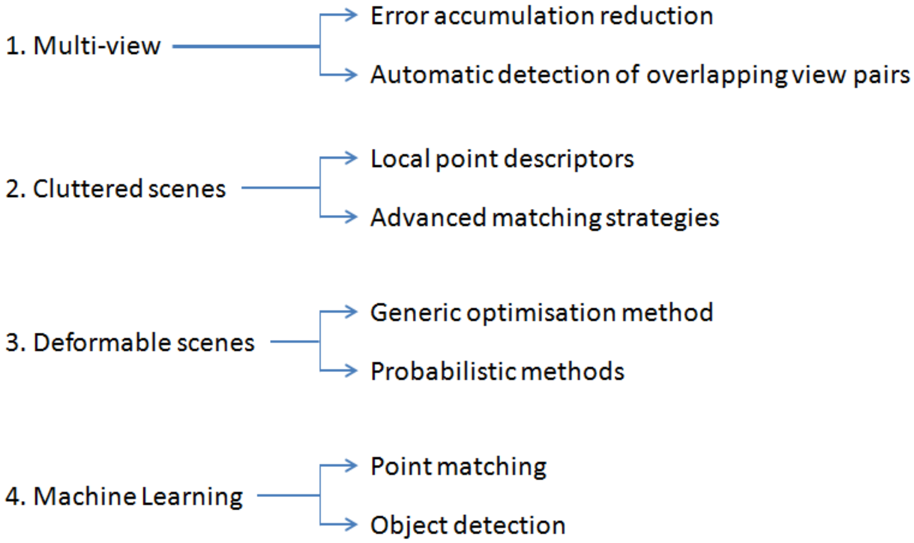


Fig. 7 A taxonomy of advanced registration techniques.

the interdependences 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 [26]. In [68] a method is presented that first aligns the scans pairwise with each other and then uses the pairwise alignments as constraints in a multi-view step. The aim is to evenly distribute the pairwise registration error, but the method itself is still based on pairwise alignments. In [16] 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, [88] 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.

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 [40] 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 [52] the pre-alignment is performed by extracting global features from each view, namely extended Gaussian images. Conversely, in [46] 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).

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. However, in cluttered scenes, an object of interest may be made of a small subset of the entire view. This makes the registration problem more challenging. Figure 8 shows two examples of highly cluttered scenes: an entire square[2] and a scene composed of several mechanical objects.

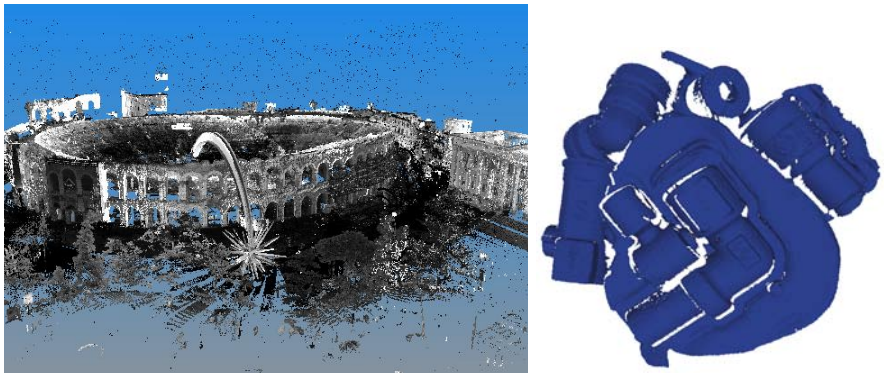


Fig. 8 Example of large scan acquisition (left) and scene with multiple mechanical objects (right).

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.

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

Point signatures

This approach is similar to local approaches for pre-alignment. Here, due to the cluttered scene, the challenge comes from the fact that the neighborhood of one point of an object can cover part of other objects. Therefore, the descriptor may become useless. In [53] a descriptor 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 [3] 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 are very few within cluttered scenes standard methods for outlier rejection are not useful but more complex matching algorithm can be exploited. In [53] descriptors are stored using a hash table that can be efficiently looked up at the matching phase by geometric hashing algorithm. In [3] matching is performed in hierarchical fashion by using the hierarchy induced from the definition of point-descriptor. In [28] 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.

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. 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 [55, 48, 20, 21]. Roughly speaking we can emphasise 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 [48, 23, 20, 21]. Moreover, other unknowns can be used to model further information like the overlapping area, the reliability of correspondences, the smoothness constraint and so on [48]. Examples of transformation models which have been introduced for surface deformations are *(i)* affine transforms applied to nodes uniformly sampled from the range images [48], *(ii)* rigid transforms on patches automatically extracted from the surface [20], *(iii)* Thin-Plate Splines (TPS) [23, 73], or *(iv)* linear blend skinning model (LBS) [21]. The error function can be optimized by the Levenberg-Marquardt Algorithm [48], GraphCuts [20], or Expectation-Maximization (EM) [23, 21, 58]. In [39] 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 [11, 12]. Also in this case the authors use TPS to represent the deformable warp between a pair of views, that they estimate through hierarchical ICP [73].

Probabilistic methods

Using probabilistic methods the uncertainty on the correct surface transformation can be addressed by adopting maximum likelihood estimation [43, 58, 92, 2, 27, 38]. Probabilistic approaches are based on modeling each of the point sets by a kernel density function [90]. 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 [2]. In [43], the authors propose a correlation-based approach [90] 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 [92], 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 [58], 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 [27] to track the full hand motion. A stereo set-up 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 [38] 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.

3.4 Machine learning techniques

Recently, advanced machine learning techniques have been exploited to improve registration algorithms [85, 1, 41, 56, 34]. 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.

Improving the matching

In these approaches the emphasis is on the effectiveness of the correspondence computation. In [85] 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 [1] 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.

Object detection

A new class of methods is emerging from employing machine learning techniques for detecting specific classes of objects on large scenes [41, 56, 34]. Several works have been done for the 2D domain, but its extension to 3D scenes is not trivial. In [41] 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 [56] 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 [34] different objects are simultaneously detected by hierarchical segmentation of point clouds. Indeed, clusters of points are classified using standard learning by example classifiers.

4 Quantitative Performance evaluation

In this section we report some experiments using pairwise registration in order to show in practice how basic registration techniques work. Two objects are evaluated: `Foot` and `Frog` for which two views are available for pairwise registration [3]. In Figure 9 we show the two objects, the overlapping region of both the objects in the two views is quite large, i.e., around 80%. The `Foot` model has approximately 1500 points, while `Frog` is composed of around 5000 points for each view. The two views have been acquired separately by two scans and are already aligned by the authors of [76], as shown in Figure 9 (right). In order to evaluate the robustness of pairwise registration methods against the initial pose variation we define 30 poses by generating random angles sampled from a uniform distribution between 0° and 10° (for the three Euler's angles) and random translations (for the three translation components t_x, t_y, t_z) such that the translation is 5% of the object main diagonal. Indeed, such transformations are considered as ground truth. We evaluate the performance of three methods: i) Besl's ICP [6] (`Besl`), which is described in Section 2.2, ii) Chen and Medioni ICP [22],(`Chen`) which introduces the point-to-plane distance, and iii) Picky ICP proposed by Zinsser et. al [97] (`Picky`) which implements a combination of ICP variations described in Section 2.3. A hierarchical sampling strategie is introduced to improve the speed, and a thresholding approach on the residual distribution is employed. More specifically, a treshould is define as $TH = \mu + 2.5\sigma$, where μ=$mean(\{e_i\})$, and σ=$std(\{e_i\})$. According to the basic ICP algorithm described in Section 2.2 the threshold of Step 4 is set as 0.00001 but in practice in the most of the evaluated experiments the algorithm stops because they reach the maximum number of iterations. Therefore, we define two settings with maximum number of iterations 15 and 50 respectively in order to evaluate the speed of convergence for the analysed cases. In Figure 10 computational efficiency is evaluated. Experiments were carried out on an entry level laptop at 1.66 Ghz with 4Gb. The code is in Matlab. The three methods are shown in red (`Besl`), blue (`Chen`), and green (`Picky`). In general, the best results were obtained by the `Picky` algorithm. Moreover, due to the higher computational cost of point-to-plane distance the

[3] Experimental material is based on the survey paper [76]. Objects and code are availabe at `http://eia.udg.es/cmatabos/research.htm`

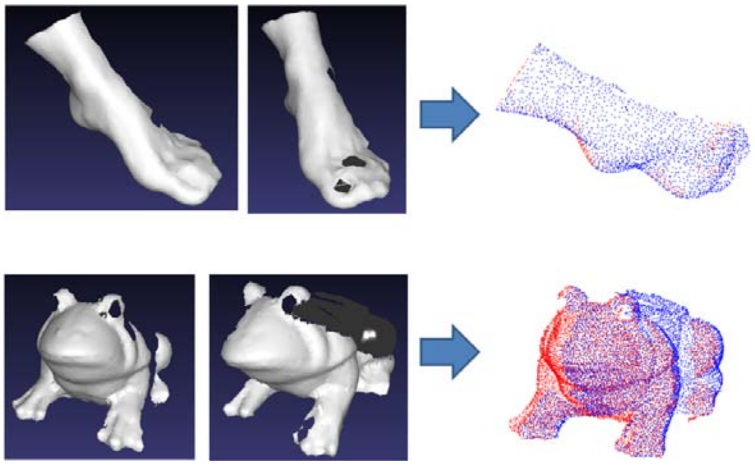


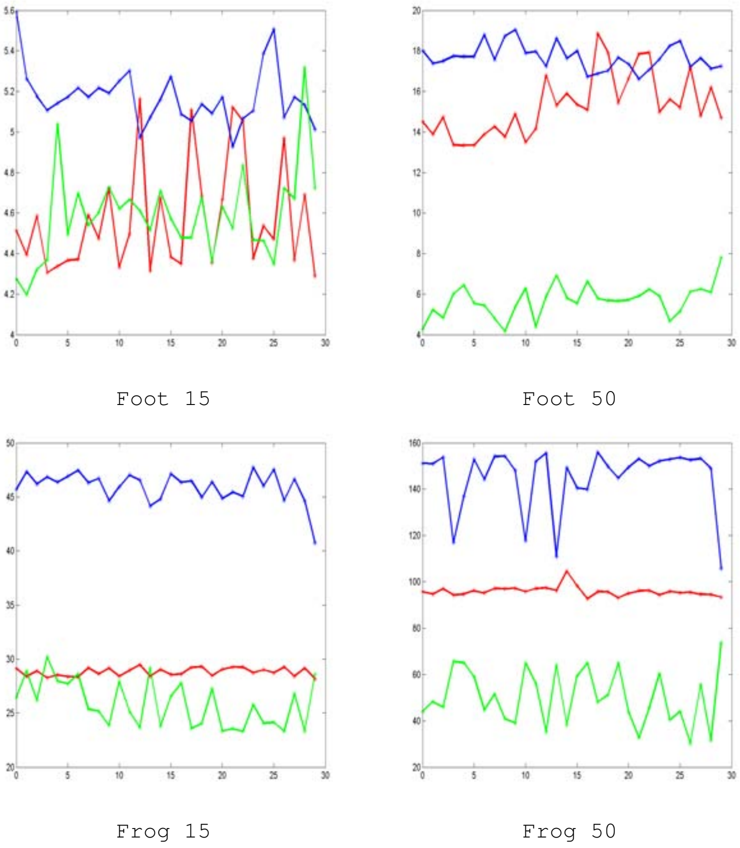
Fig. 9 Two views of Foot (Top) and Frog (Bottom). Separated views (left) and aligned views(right). The portion of overlap is large for both the objects.

Chen method is the slowest. Note that a drastic reduction of the registration speed is observed when the number of points increase from Foot to Frog.

In Figure 11 the accuracy of registration is evaluated. In general, especially when the ground truth is not available, a good accuracy evaluation criterion is the *Mean Squared Error* (MSE) (i.e, $mean\{e_i\}$). Note that the MSE error is not comparable between the three methods since they compute different distance measures, and in Picky the error generated by outliers is not considered. Note that for both the objects MSE error of Besl does not improve when the number of iterations increases. Conversely, in Picky the benefit of a higher number of iterations is observed and the most of the registrations are able to reach convergence. In order to get a direct comparison between the methods we use the ground truth.

transformations which are available since we have generated them. Indeed, we evaluate the Rotation and the Translation error as shown in Figures 12 and 13 respectively. Rotation error is computed as the mean of the difference between the observed Euler's angles of the estimated transform and the ground truth. Angles are in radians. Translation error is the norm of the difference between the two translation vectors.

Table 1 summarizes the experiments by showing the mean of the evaluated measures. We observe better accuracy by using Picky. Note also that in general Chen outperforms Besl. Finally, we also show in Table 1 the number of failures (or divergences) because a local minimum is reached. We show that in more than half of the cases the convergence is reached within 15 iterations. Again the best results are shown with Picky since no failures are observed after 50 iterations and it always converges after 15 iterations with Foot. We observed also that the cases of failures of Besl and Chen methods are related to the strongest variations of starting pose thus confirming the importance of a good initialization procedure.



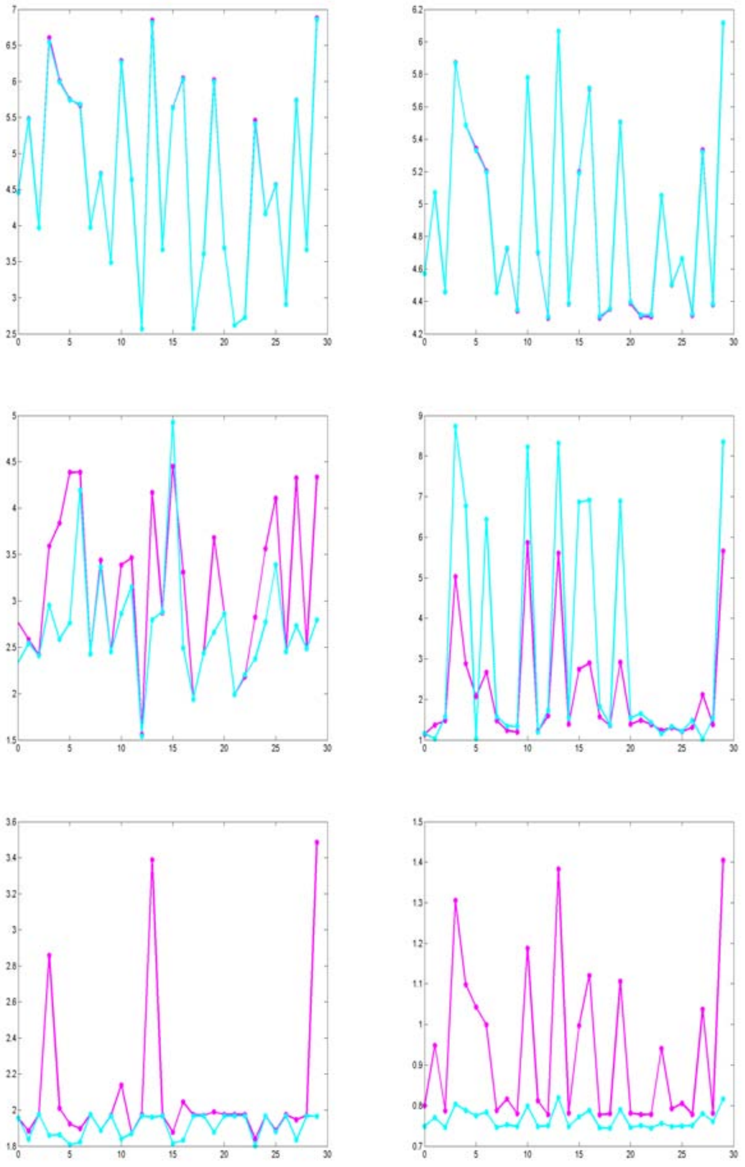
`Foot 15` `Foot 50`

`Frog 15` `Frog 50`

Fig. 10 Computational time evaluation. Time (sec.) employed for each pairwise registration. The three methods are shown in red (`Besl`), blue (`Chen`), and green (`Picky`).

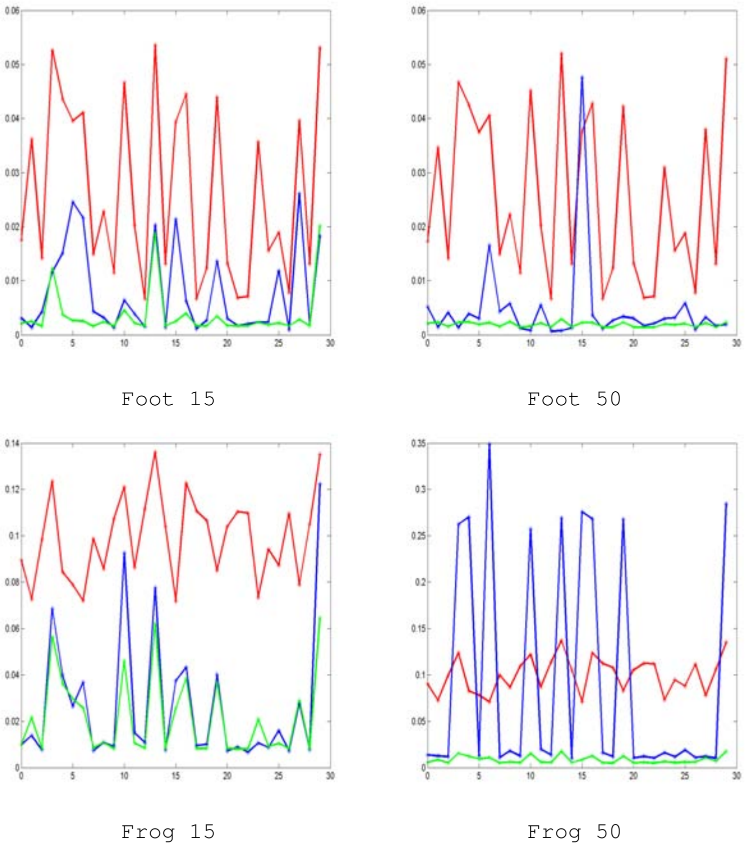
Exp.	**MSE error**(mm.)	**Rot-error (rad.)**	**Transl-error** (mm.)	**# Divergences**	**Time (sec.)**
Besl Foot 15	4.681	0.026	0.922	13/30	4.568
Chen Foot 15	3.155	0.007	0.593	15/30	5.173
Picky Foot 15	2.084	0.003	0.195	6/30	0.593
Besl Foot 50	4.672	0.025	0.853	13/30	15.309
Chen Foot 50	2.725	0.004	0.601	5/30	17.680
Picky Foot 50	1.910	0.001	0.104	0/30	5.683
Besl Frog 15	4.905	0.099	2.936	14/30	28.811
Chen Frog 15	2.203	0.026	0.976	10/30	45.917
Picky Frog 15	0.932	0.021	0.411	9/30	25.850
Besl Frog 50	4.906	0.099	2.939	5/30	95.882
Chen Frog 50	3.217	0.092	2.608	9/30	145.028
Picky Frog 50	0.765	0.008	0.069	0/30	50.292

Table 1 Summary of performance evaluation. The mean of the 30 pairwise registrations for each evaluated measure is shown.



Foot Frog

Fig. 11 MSE evaluation for `Besl` (top), `Chen` (middle), and `Picky` (bottom). The cases employing 15 iterations are shown in magenta, while cases employing 30 iterations are shown in cyan.



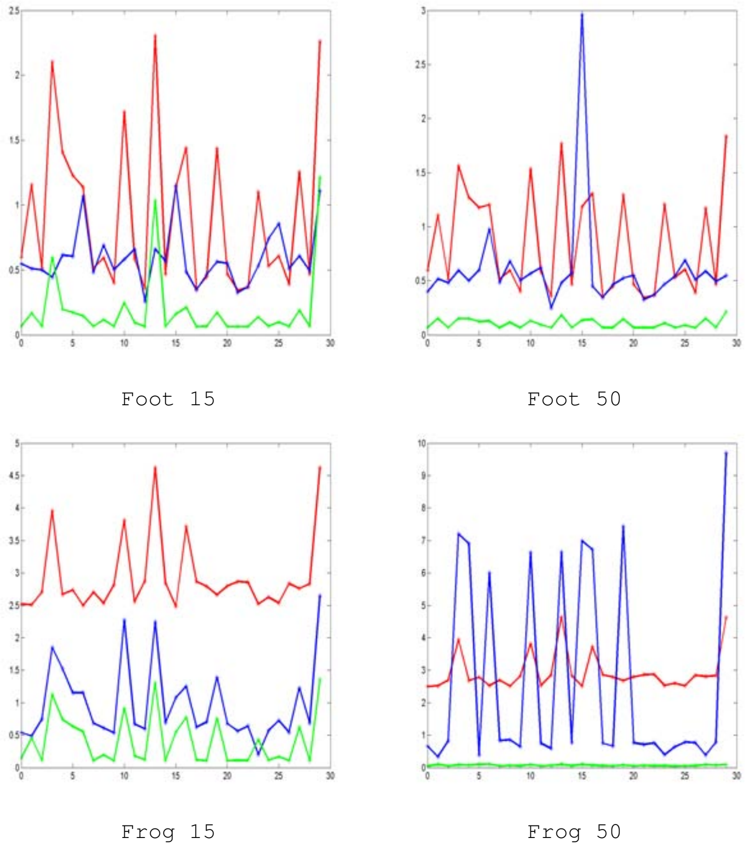
`Foot 15` `Foot 50`

`Frog 15` `Frog 50`

Fig. 12 Rotation error. The three methods are shown in red (`Besl`), blue (`Chen`), and green (`Picky`). Note that peaks correspond to failures of the pairwise registration procedure.

5 Case Study 1: Pairwise Alignment with Outliers 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 [16, 37]. 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 [96] and consists of considering the residuals of two fully overlapping sets as an approximation of a Gaussian distribution. Non-overlapping points can be detected



Foot 15

Foot 50

Frog 15

Frog 50

Fig. 13 Translation error. The three methods are shown in red (`Besl`), blue (`Chen`), and green (`Picky`). Note that peaks correspond to failures of the pairwise registration procedure.

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 $\mathbf{e} = [e_1, \ldots, e_{N_d}]$, the *Median Absolute Deviation* (MAD) is defined as:

$$MAD = med(|e_i - location|), \tag{5}$$

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

$$|e_i - location| < k \cdot MAD. \tag{6}$$

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 equation 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 $med(\mathbf{e})$.
3. Compute *MAD* according to equation 5.
4. For each residual error e_i $(i = 1, \ldots, N_d)$:

   a. If e_i satisfies equation 6 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 section 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.

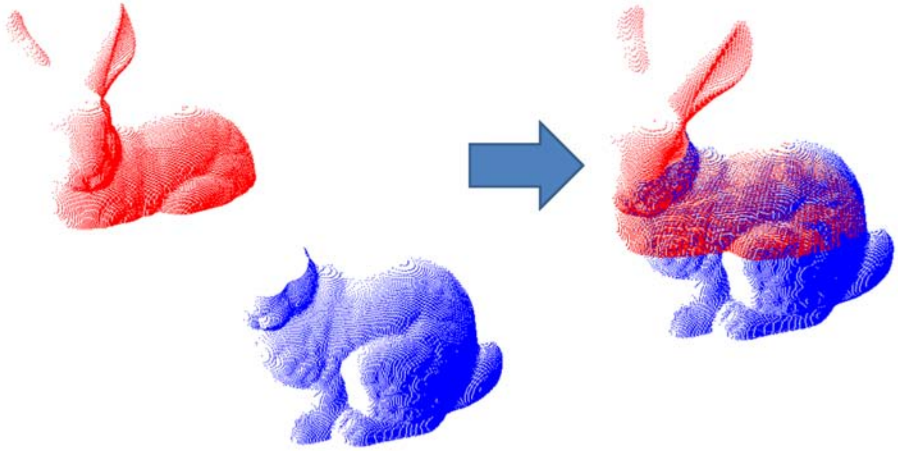


Fig. 14 Registration with robust outliers rejection. Two views at starting pose (left) and after registration (right). Note that the overlap area is quite restricted.

In figure 14 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 $\#(D \cap M) = 4000$. 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 15 shows the distribution of residual errors after `X84-ICP` registration. Note that most of the residuals are concentrated around zero. It is confirmed that the behavior of the early part of the distribution is similar to a Gaussian [96]. 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[4]. In figure 15 (right) points of the data-view are colored differently between inliers and outliers. Note that non-overlapping parts are correctly registered.

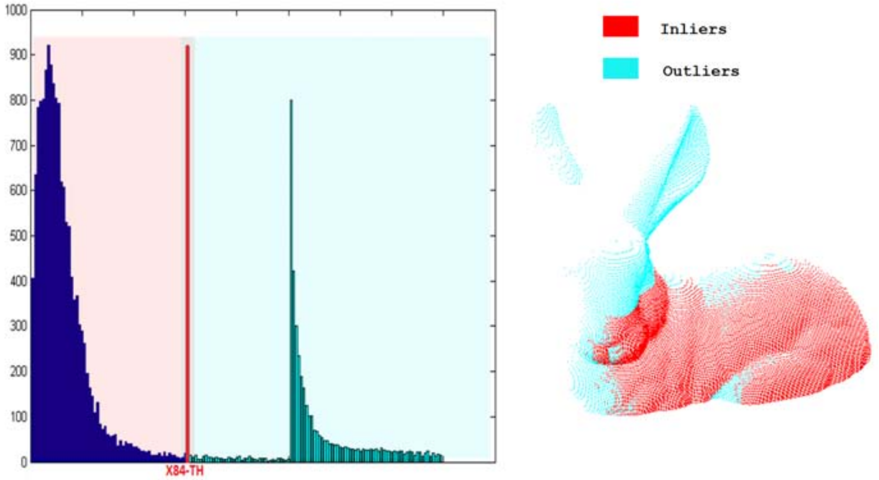


Fig. 15 Automatic residuals thresholding. From the distribution of residuals the threshold is estimated according to the X84 rule. Points under the threshold are inliers (red), while outliers are over the threshold (blue). Outliers are points in non-overlapping areas.

Method	**Rot-error** (rad.)	**Transl-error** (mm.)	**# Overlap. points**	**# Iterations**	**Time** (sec.)
Besl	0.22345	1.2636	10000	20	370
Picky	0.10918	0.9985	9534	28	76
X84-ICP	0.06351	0.4177	4582	21	383
Ground Truth	-	-	4000	-	-

Table 2 X-84 performance evaluations. Rotation and translation errors are reported.

[4] In order to visualize the peak the second part of the histogram has been quantized with wider intervals.

Table 2 summarizes the performance of `X84-ICP` in comparison with `Besl` and `Picky`. 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 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. We highlight that although X84-ICP performs well in this experiment, in more general cases if the number of outliers is greater than 50% of the residual distribution the *X84* rule is likely to fail.

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 [30] was proposed in order to minimize the alignment error by employing a nonlinear optimization procedure. The advantage of the LM-ICP is the versatility in the definition of the optimization function in order to take into account of several aspects of the registration such as, the outlier rejection and the speed.

6.1 The LM-ICP Method

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

$$E(\mathbf{a}) = \sum_{i=1}^{N_d} (e_i(\mathbf{a}))^2, \qquad e_i(\mathbf{a}) = \|\mathbf{R}\mathbf{d}_i + \mathbf{t} - \mathbf{m}_j\|. \tag{7}$$

Defining the residual vector as:

$$\mathbf{e}(\mathbf{a}) = \{e_i(\mathbf{a})\}_{i=1}^{N_d}, \tag{8}$$

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

The Levenberg-Marquardt algorithm combines gradient-descent and Gauss-Newton. The goal of 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:

$$E(\mathbf{a}+\mathbf{x}) = E(\mathbf{a}) + (\nabla E(\mathbf{a}) \cdot \mathbf{x}) + \frac{1}{2!}((\nabla^2 E(\mathbf{a}) \cdot \mathbf{x}) \cdot \mathbf{x}) + h.o.t. \tag{9}$$

This is rewritten in terms of **e** as:

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

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

$$E(\mathbf{a}+\mathbf{x}) \approx \mathbf{e}^\mathsf{T}\mathbf{e} + \mathbf{x}^\mathsf{T}\mathsf{J}^\mathsf{T}\mathbf{e} + \mathbf{x}^\mathsf{T}\mathsf{J}^\mathsf{T}\mathsf{J}\mathbf{x}. \tag{10}$$

Differentiating with respect to **x** and nullifying yields:

$$\nabla_\mathbf{x} E(\mathbf{a}+\mathbf{x}) = \mathsf{J}^\mathsf{T}\mathbf{e} + \mathsf{J}^\mathsf{T}\mathsf{J}\mathbf{x} = 0, \tag{11}$$

and gives the Gauss-Newton update:

$$\mathbf{x}_{GN} = -(\mathsf{J}^\mathsf{T}\mathsf{J})^{-1}\mathsf{J}^\mathsf{T}\mathbf{e}. \tag{12}$$

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:

$$\mathbf{x}_{GD} = -\lambda^{-1}\mathsf{J}^\mathsf{T}\mathbf{e}, \tag{13}$$

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:

$$\mathbf{x}_{LM} = -(\mathsf{J}^\mathsf{T}\mathsf{J} + \lambda I)^{-1}\mathsf{J}^\mathsf{T}\mathbf{e}. \tag{14}$$

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 Squared (IRLS)-

like reweighting procedure. For instance, the following robust functions can be used:

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

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 [30] a more effective solution was proposed based on the *distance transform* which also drastically improves the computational efficiency. The distance transform is defined as:

$$D_\varepsilon(\mathbf{x}) = \min_j \varepsilon^2(\|\mathbf{m}_j - \mathbf{x}\|), \tag{15}$$

where $\mathbf{x} \in X$ and X is a discrete grid representing the volume which encloses the model-view 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[5]. 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 $\|\mathbf{x}\| \to \varepsilon^2(\|\mathbf{x}\|)$ is monotonic, we obtain that $D_\varepsilon(\mathbf{x}) = \varepsilon^2(\|D(\mathbf{x})\|)$, so existing algorithms to compute D may be used to compute D_ε, without requiring knowledge of the form of ε.

By combining equation (7) with equation (15) the new formulation of the registration problem becomes:

$$E(\mathbf{a}) = \sum_{i=1}^{N_d} D_\varepsilon(T(\mathbf{a}, \mathbf{d}_i)). \tag{16}$$

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, $\nabla_\mathbf{x} D_\varepsilon = [\frac{\partial D_\varepsilon}{\partial_x}, \frac{\partial D_\varepsilon}{\partial_y}, \frac{\partial D_\varepsilon}{\partial_z}]$ is computed by defining $\frac{\partial D_\varepsilon(x,y,z)}{\partial_x} = \frac{D_\varepsilon(x+1,y,z) - D_\varepsilon(x-1,y,z)}{2}$, $\frac{\partial D_\varepsilon(x,y,z)}{\partial_y} = \frac{D_\varepsilon(x,y+1,z) - D_\varepsilon(x,y-1,z)}{2}$, and $\frac{\partial D_\varepsilon(x,y,z)}{\partial_z} = \frac{D_\varepsilon(x,y,z+1) - D_\varepsilon(x,y,z-1)}{2}$. In practice, $\nabla_\mathbf{x} D_\varepsilon$ remains constant through the minimization, and we get:

$$\nabla_\mathbf{a} E(\mathbf{a}) = \sum_{i=1}^{N_d} \nabla_\mathbf{x} D_\varepsilon(T(\mathbf{a}, \mathbf{d}_i)) \nabla_\mathbf{a}^\top T(\mathbf{a}, \mathbf{d}_i). \tag{17}$$

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

Note that the computation of $\nabla_{\mathbf{a}}^{\top} T(\mathbf{a},\mathbf{d}_i)$ depends on the rigid transformation parametrization being used. In [30] the author proposed to 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 $\mathsf{J}_{i,j} = (\nabla_{\mathbf{x}} D_{\varepsilon}(T(\mathbf{a},\mathbf{d}_i)) \cdot \nabla_{a_j}^{\top} T(\mathbf{a},\mathbf{d}_i))$, where $\nabla_{a_j} T(\mathbf{a},\mathbf{d}_i) = [\frac{\partial T_x(\mathbf{a},\mathbf{d}_i)}{\partial a_j}, \frac{\partial T_y(\mathbf{a},\mathbf{d}_i)}{\partial a_j}, \frac{\partial T_z(\mathbf{a},\mathbf{d}_i)}{\partial a_j}]$.

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 [93]. Let $\mathbf{d}$ be the point on which the rotation must be applied. To this aim such point must be represented onto the quaternion space by leading to $\mathbf{r} = [0,\mathbf{d}]$. Therefore the rotated point is obtained by:

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

By solving the moltiplication onto the quaternion space[6] 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:

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

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

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

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

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

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

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

$$\nabla_{\mathbf{a}}^{\top} T(\mathbf{a}, \mathbf{d}) = \begin{pmatrix} \frac{\partial T_x}{\partial s} & \frac{\partial T_x}{\partial v_x} & \frac{\partial T_x}{\partial v_y} & \frac{\partial T_x}{\partial v_z} & \frac{\partial T_x}{\partial t_x} & \frac{\partial T_x}{\partial t_y} & \frac{\partial T_x}{\partial t_z} \\ \frac{\partial T_y}{\partial s} & \frac{\partial T_y}{\partial v_x} & \frac{\partial T_y}{\partial v_y} & \frac{\partial T_y}{\partial v_z} & \frac{\partial T_y}{\partial t_x} & \frac{\partial T_y}{\partial t_y} & \frac{\partial T_y}{\partial t_z} \\ \frac{\partial T_z}{\partial s} & \frac{\partial T_z}{\partial v_x} & \frac{\partial T_z}{\partial v_y} & \frac{\partial T_z}{\partial v_z} & \frac{\partial T_z}{\partial t_x} & \frac{\partial T_z}{\partial t_y} & \frac{\partial T_z}{\partial t_z} \end{pmatrix} \tag{18}$$

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:

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

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

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 $D_{\varepsilon}(\mathbf{x})$,
3. set $\mathbf{a}_k \leftarrow \mathbf{a}_0$,
4. compute $\mathbf{e}_k = \mathbf{e}(\mathbf{a}_k)$,
5. compute J,

```
6. repeat
    a. compute update $\mathbf{a}_{k+1} = \mathbf{a}_k - (\mathsf{J}^T\mathsf{J} + \lambda I)^{-1}\mathsf{J}^T)\mathbf{e}_k$
    b. compute $\Delta E = E(\mathbf{a}_{k+1}) - E(\mathbf{a}_k)$
    c. If $\Delta E > 0$ then $\lambda = 10\lambda$, goto a. else $\lambda = \frac{1}{10}\lambda$, goto 4.
7. If $\|\mathbf{e}_k\| > \nu$ goto 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 J. In particular, the distance transform $D_\varepsilon(\mathbf{x})$, enables an improvement of the computational efficiency of the error computation and makes the computation of Jacobian feasible. The starting value $\mathbf{a}_0$ can be estimated by employing some of the techniques described in section 2.3.1.

6.5 Results and discussion

Figure 16 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 $1s$ for an LM iteration. A total of 20 iterations has been run to reach converge. 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 $30s$. This confirms that a drastic improvement of the speed is observed with LM-ICP in comparison with basic ICP. Note that a crucial parameter is the grid size. It trades off between computational efficiency and memory space. Moreover it requires that 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 [30]. 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 [30] for more details).

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

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) [18]. DLM-ICP extends the LM-ICP introduced in section 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

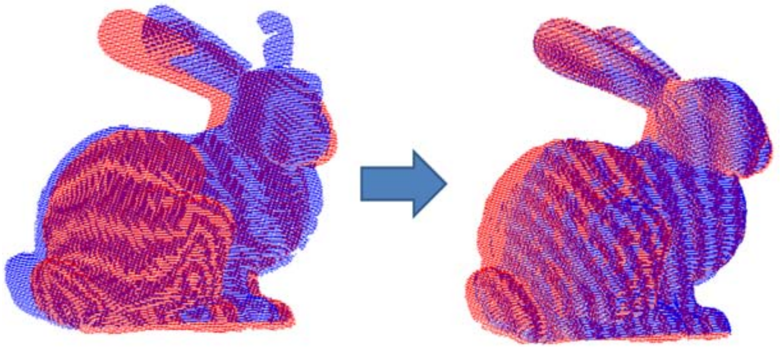


Fig. 16 LM-ICP. The starting pose (left) and merged views after registration (right).

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 [30] described in section 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 [30], it explicitly embeds a min operator, thus avoiding the traditional two steps in ICP-like algorithms through 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 to slide 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 tractable and fast the estimation of dense deformation fields.

7.1 Surface Representation

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

$$D_i = \begin{pmatrix} d^x_{i,1} & d^y_{i,1} & d^z_{i,1} \\ \vdots & \vdots & \vdots \\ d^x_{i,l_i} & d^y_{i,l_i} & d^z_{i,l_i} \end{pmatrix}.$$

The unknown model $\mathbf{a} = M$ 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 M_i as:

$$M_i = \begin{pmatrix} m^x_{i,1} & m^y_{i,1} & m^z_{i,1} \\ \vdots & \vdots & \vdots \\ m^x_{i,N_m} & m^y_{i,N_m} & m^z_{i,N_m} \end{pmatrix}.$$

In practice, the number of data points is much larger than the number of model points, i.e. $l_i \gg N_m$. 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, differently than section 6, the registration is from model-points to data-points.

7.2 Cost Function

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

$$E(M) = E_g(M) + \lambda_b E_b(M) + \lambda_s E_s(M) + \lambda_t E_t(M) + \lambda_x E_x(M), \tag{19}$$

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

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 ones respectively account for *spatial smoothness* and *temporal smoothness* E_s. 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 the five terms are explained below in details.

Data term: global surface attraction

This term globally attracts the model to the data points in a closest point manner [75]. Denoting B_M and B_D the sets of boundary points in the model and in the data, we get the following data term, integrating the model to data points matching step:

$$\sum_{\mathbf{m}\in M\backslash B_M} \min_{\mathbf{d}\in D\backslash B_D} \parallel \mathbf{d}-\mathbf{m} \parallel^2, \tag{20}$$

where $\mathbf{d}$ and $\mathbf{m}$ are $3-$vectors respectively representing a data and a model point. 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 M.

An *outliers rejection* strategy is introduced by defining a robust function ε. Here, the $X84$ rule is employed [16]. Therefore, (20) is modified so as to get the following robustified data term:

$$E_g(M) = \sum_{\mathbf{m}\in M\backslash E_M} \varepsilon\left(\min_{\mathbf{d}\in D\backslash B_D} \parallel \mathbf{d}-\mathbf{m} \parallel^2\right). \tag{21}$$

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 (21) except that the sum and min operators are over the boundary points:

$$E_b(M) = \sum_{\mathbf{m}\in B_M} \varepsilon\left(\min_{\mathbf{d}\in B_D} \parallel \mathbf{d}-\mathbf{m} \parallel^2\right). \tag{22}$$

Note that the boundaries can be computed by combining edge detection techniques with morphological operators[7]. 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 that boundaries of the model and the observed surface coincide.

[7] The 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. On stereo sensors they can be estimated on one of the two optical views.

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 [36], the model M is a displacement field parameterized[8] by (u,v) with $u = [1 \dots R]$ and $v = [1 \dots C]$, i.e., $M(u,v) = [M^x(u,v), M^y(u,v), M^z(u,v)]$. The spatial smoothness term can thus be taken as the surface bending energy:

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

Using a finite difference approximation for the first and second derivatives [67], the bending energy can be expressed in the discrete form as a quadratic function of M. More specifically, the derivatives $\frac{\partial M^x}{\partial u}$ at a point (u,v) is discretely approximated as $\frac{\partial M^x(u,v)}{\partial u} = M^x(u+1,v) - M^x(u-1,v)$. This can be conveniently represented by a constant $N_m \times N_m$ matrix C_u such that $\nabla_u M^x = C_u \cdot \mathrm{vect}(M^x)$, where $\mathrm{vect}(M^x)$ is the vectorization operator which rearranges matrix M^x to a vector. A similar matrix C_v can be computed with respect to v. Indeed, the second derivatives are computed using Hessian operator matrices, namely C_{uu}, C_{uv}, C_{vv}. The surface bending energy can be expressed in discrete form by defining:

$$E_s^x = \mathrm{vect}(M^x)^\top (C_{uu}^\top C_{uu} + 2C_{uv}^\top C_{uv} + C_{vv}^\top C_{vv}) \mathrm{vect}(M^x),$$

and by computing:

$$E_s(M) = E_s^x(M^x) + E_s^y(M^y) + E_s^z(M^z),$$

which can be further expressed in matrix form as follows:

$$E_s(M) = \mathrm{vect}(M)^\top \mathscr{K} \mathrm{vect}(M), \tag{23}$$

where $\mathscr{K}$ 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, M and $\tilde{M}$:

$$E_t(M) = \parallel M - \tilde{M} \parallel^2 . \tag{24}$$

This makes the surface deformation smooth over time and can be used within a sequential processing approach. It is obviously not used on the first frame of the sequence.

[8] Recall that the model points lie on a grid.

Penalty term: non-extensibility

This term discourages surface stretching. It favors the mesh vertices to preserve their distance with their local neighborhood [77]:

$$E_X(M) = \sum_{\mathbf{m}\in M} \sum_{\mathbf{k}\in\mathcal{N}(\mathbf{m})} \left(\| \mathbf{m} - \mathbf{k} \|^2 - L_{m,k}^2\right)^2, \tag{25}$$

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 $\mathbf{m}$.

7.3 Minimization Procedure

The DLM-ICP cost function (19) is a sum of squared residuals nonlinearly depending on the unknowns in M. Therefore, as in section 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:

$$J^\top = \begin{pmatrix} J_d^\top & J_b^\top & J_s^\top & J_t^\top & J_x^\top \end{pmatrix}. \tag{26}$$

where $J_d^{N_m\times 3N_m}$, $J_b^{N_B\times 3N_m}$, $J_s^{3N_m\times 3N_m}$, $J_t^{N_m\times 3N_m}$, $J_x^{\xi\times 3N_m}$, are related to the global attraction, boundary attraction, spatial smoothness, temporal smoothness and non-extensibility terms respectively, and $\xi = \mathtt{size}(\mathcal{N}(\mathtt{M}))$. In particular, the Jacobians of global and boundary attraction terms are estimated by finite differences through distance transform as described in section 6.

Note that in this case, since the Hessian matrix[9] $H = J^\top J + \lambda I$ 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 J is very sparse. Thus, it uses the sparsity to speed up each iteration using the technique in [65]. In particular, a sparse Cholesky factorization package can be used as in the Matlab 'mldivide' function.

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 D_i
   a. Extract data boundary B_D

[9] We use 'Hessian matrix' for the damped Gauss-Newton approximation to the true Hessian matrix.

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. Goto 3.

Step 3.c is described in section 6.4. Here, the unknown is $\mathbf{a} = M_i$, the error function $E(M_i)$ is defined by equation (19), and the Jacobian J is defined by equation (26).

7.5 Experiments

In the following experiment the sensor is a real-time passive-stereo system[10]. The sensor acquires the 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 17 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 17 also shows the image-boundary extracted by combining binary image segmentation method with 2D morphological operators and depicts the 3D data (i.e., the selected point cloud and 3D boundary).

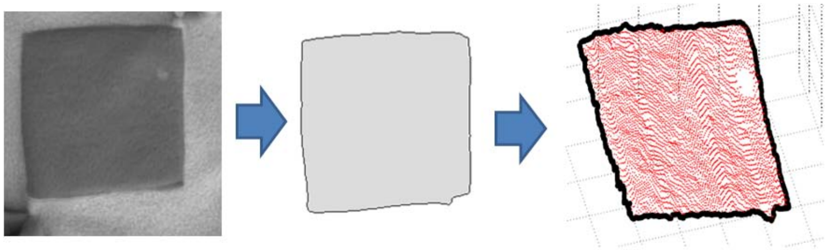


Fig. 17 Data acquisition: intensity image of the blanket (left), image-boundary (center), and the 3D point cloud (right).

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 fit on 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. 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 result accuracy. In this experiment the volume is divided into $36 \times 36 \times 18$ voxels. Figure 18 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

[10] Data courtesy of eVS (http://www.evsys.net).

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 [18].

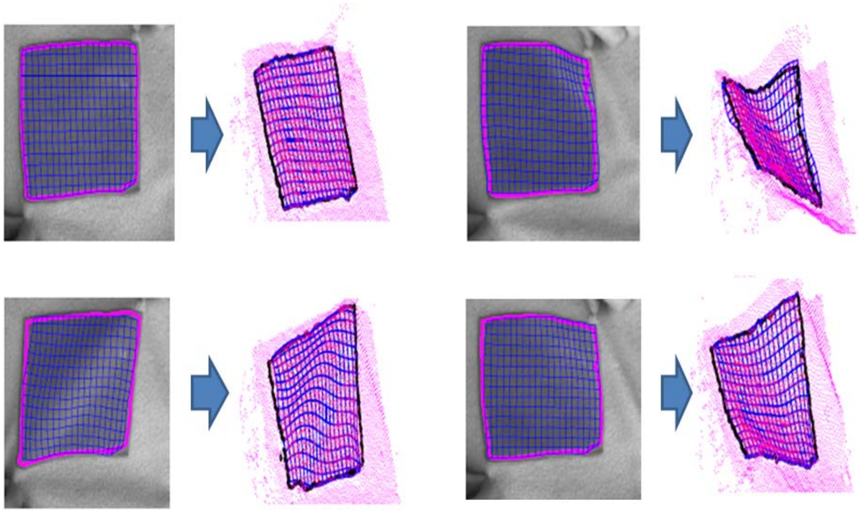


Fig. 18 Blanket sequence: 4 selected frames. For each frame the 2D intensity and the 3D data is visualized. The grid models are shown in the 3D space as well as their projection in the 2D image.

8 Challenges and Future Directions

In general, the new challenges of registration methods arise from the advances of new acquisition procedures. Structure and motion reconstruction techniques are now available to provide accurate sparse or dense reconstructed scenes from 2D images. Large-scale scanners are also able to acquire wide scenes. The registration of data coming from these procedures are challenging due to strong clutter and occlusions. Moreover, as observed before, the object to be registered may be very small with respect to the whole scene. An important issue is the local scale estimation of scene sub-parts. On the other hand texture or color information can be also acquired by the sensor. Therefore, registration can be improved by integrating effectively these additional cues. Another promising direction is the use of machine learning techniques as those described before. In particular, new techniques can be exploited, inspired from similar issues already addressed for the 2D domain like face, car or pedestrian detection techniques. Improvements can be achieved by integrating 3D scans and 2D images.

Other problems need to be addressed when real-time scanners are used. In this scenario, objects can move (change their pose) or deform. Therefore, deformable registration techniques should be employed. In particular, all the advances on isometry-invariant point correspondence computation can improve the deformable registration. Other issues are coming from the explosion of data collection. For instance, from real-time scanners a large amount of data can be acquired. In order to avoid exhaustive search some more effective matching strategies can be exploited. Feature based techniques are useful to this aim. In particular, feature point detection and description can reduce drastically the number of analyzed points. Also hierarchical techniques are needed to reduce the search space. Finally, to design a proper surface deformation transform, deformable registration methods can be inspired from 3D animation techniques.

9 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 section 2.3 are sufficient to obtain a 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 section 3. In order to give some examples of registration algorithms three case studied 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. Finally, 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.

New challenging scenarios can be addressed as described in section 8 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.

10 Further Reading

In order to get a more comprehensive overview of 3D registration methods the reader can refer to recent surveys [75, 54, 45, 76]. In [75], 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 [54]. In [76] Salvi et al. proposed an extensive experimental comparison amongst different 3D pairwise registration methods. They evaluated the accuracy of the results for both coarse and fine registration. More recently, Kaick et al. [45] proposed a survey on shape correspondence estimation by extensively reporting and discussing interesting methods for deforming scenarios.

The reader interested in getting in-depth details on the theoretical evaluation of registration convergence should refer to Pottmann et al.'s work [32, 66]. Convergence is discussed also by Ezra et al. [29] who provided lower and upper bounds on the number of ICP iterations. One of these methods [83] defines a new registration metric called the 'surface interpenetration measure'. This is in contrast to the mean square error (MSE) employed by classical ICP and the authors claim that this is more effective when attempting to achieve precise alignments. Finally, we claimed already that most of the registration techniques are based on ICP algorithm. Alternative methods can be considered such as those based on Genetic Algorithms [83, 50, 49].

11 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 What advantages does LM-ICP have over classical ICP?

12 Exercises

1. Given two partial views very close to each other and an implementation of ICP[11] try to register the views by gradually moving away 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 among *(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.[12]
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 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. 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)
2. Anguelov, D., Srinivasan, P., Pang, H.C., Koller D., T., S. Davis, J.: The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In: Neural Information Processing Systems Conference (2004)

[11] A Matlab implementation can be found here http://www.csse.uwa.edu.au/ajmal/code.html.

[12] A Matlab implementation is available here: http://www.robots.ox.ac.uk/ awf/lmicp.

3. Bariya, P., Nishino, K.: Scale-hierarchical 3d object recognition in cluttered scenes. In: International Conference on Computer Vision and Pattern Recognition (2010)
4. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(4), 509–522 (2002)
5. Bernardini, F., Rushmeier, H.: The 3D model acquisition pipeline. Computer Graphics Forum **21**(2), 149–172 (2002)
6. Besl, P., McKay, H.: A method for registration of 3-D shapes. IEEE Transactions on pattern analysis and machine intelligence **14**(2), 239–256 (1992)
7. Blais, G., Levine, M.: Registering multiview range data to create 3d computer objects. IEEE Trans. Pattern Anal. Mach. Intell. **17**(8) (1995)
8. Bowyer, K.W., Chang, K., Flynn, P.: A survey of approaches and challenges in 3d and multimodal 3d + 2d face recognition. Computer Vision Image Understanding **101**(1) (2006)
9. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Three-dimensional face recognition. International Journal of Computer Vision **64**(1), 5–30 (2005)
10. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Numerical geometry of non-rigid shapes. Springer Verlag (2008)
11. Brown, B., Rusinkiewicz, S.: Non-rigid range-scan alignment using thin-plate splines. In: Symposium on 3D Data Processing, Visualization, and Transmission (2004)
12. Brown, B., Rusinkiewicz, S.: Global non-rigid alignment of 3-D scans. ACM Transactions on Graphics (Proc. SIGGRAPH) **26**(3) (2007)
13. 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)
14. Campbell, R., Flynn, P.: A survey of free-form object representation and recognition techniques. Computer Vision and Image Understanding **81**(2), 166–210 (2001)
15. 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)
16. Castellani, U., Fusiello, A., Murino, V.: Registration of multiple acoustic range views for underwater scene reconstruction. Computer Vision and Image Understanding **87**(3), 78–89 (2002)
17. Castellani, U., Fusiello, A., Murino, V., Papaleo, L., Puppo, E., Pittore, M.: A complete system for on-line modelling of acoustic images. Image Communication Journal **20**(9-10), 832–852 (2005)
18. Castellani, U., Gay-Bellile, V., Bartoli, A.: Robust deformation capture from temporal range data for surface rendering. Computer Animation and Virtual Worlds **19**(5), 591–603 (2008)
19. Chang, M., Leymarie, F., Kimia, B.: 3d shape registration using regularized medial scaffolds. In: International Symposium on 3D Data Processing, Visualization and Transmission (2004)
20. Chang, W., Zwicker, M.: Automatic registration for articulated shapes. Computer Graphics Forum (Proceedings of SGP 2008) **27**(5), 1459–1468 (2008)
21. Chang, W., Zwicker, M.: Range scan registration using reduced deformable models. Computer Graphics Forum, to appear **28**(2), 447–456 (2009)
22. Chen, Y., Medioni, G.: Object modelling by registration of multiple range images. Image and Vision Computing **10**(3), 145–155 (1992)
23. Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. Computer Vision and Image Understanding **89**(2-3), 114–141 (2003)
24. 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)
25. 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)
26. Cunnington, S., Stoddart, A.: N-view point set registration: A comparison. In: British Machine Vision Conference (1999)
27. Dewaele, G., Devernay, F., Horaud, R.: Hand motion from 3d point trajectories and a smooth surface model. In: European Conference on Computer Vision (2004)

28. 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)
29. Ezra, E., Sharir, M., Efrat, A.: On the performance of the icp algorithm. Computational Geometry **41**(1-2), 77 – 93 (2008)
30. Fitzgibbon, A.: Robust registration of 2D and 3D point sets. Image and Vision Computing **21**(13-14), 1145 – 1153 (2003)
31. Funkhouser, T., Kazhdan, M., Min, P., Shilane, P.: Shape-based retrieval and analysis of 3d models. Communications of the ACM **48**(6), 58–64 (2005)
32. Gelfand, N., Mitra, N.J., Guibas, L.J., Pottmann, H.: Robust global registration. In: In M. Desbrun and H. Pottmann, editors, Eurographics Association, ISBN 3-905673-24-X., pp. 197–206 (2005)
33. 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)
34. Golovinskiy, A., Kim, V., Funkhouser, T.: Shape-based recognition of 3d point clouds in urban environments. In: International Conference on Computer Vision (2009)
35. Granger, S., Pennec, X.: Multi-scale em-icp: a fast and robust approach for surface registration. In: European Conference on Computer Vision (2002)
36. Gu, X., Gortler, S.J., Hoppe, H.: Geometry images. ACM Transaction on Graphics **21**(3), 355–361 (2002)
37. Hampel, F., Rousseeuw, P., Ronchetti, E., Stahel, W.: Robust Statistics: The Approach Based on Influence Functions. Wiley (1986)
38. 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)
39. Huang, Q., Adams, B., Wicke, M., Guibas, L.: Non-rigid registration under isometric deformations. Computer Graphic Forum **27**(5), 1449–1457 (2008)
40. Huber, D., Hebert, M.: Fully automatic registration of multiple 3D data sets. Image and Vision Computing **21**(7), 637–650 (2003)
41. IV, A.P., 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)
42. Jhonson, A., Kang, S.: Registration and integration of textured 3d data. Image and Vision Computing **19**(2), 135–147 (1999)
43. 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)
44. Johnson, A., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence **21**(5), 433–449 (1999)
45. van Kaick, O., Zhang, H., Hamarneh, G., Cohen-Or, D.: A survey on shape correspondence. In: In EuroGraphics: State-of-the-Art Report (2010)
46. Khoualed, S., Castellani, U., Bartoli, A.: Semantic shape context for the registration of multiple partial 3--D views. In: British Machine Vision Conference (2009)
47. Krsek, P., Pajdla, T., Hlavc, V.: Differential invariants as the base of triangulated surface registration. Computer Vision and Image Understanding **87**(1-3), 27 – 38 (2002)
48. Li, H., Sumner, R.W., Pauly, M.: Global correspondence optimization for non-rigid registration of depth scans. Computer Graphics Forum (Proc. SGP'08) **27**(5) (2008)
49. Liu, Y.: Automatic 3d free form shape matching using the graduated assignment algorithm. Pattern Recognition **38**, 1615–1631 (2005)
50. Lomonosov, E., Chetverikov, D., Ekárt, A.: Pre-registration of arbitrarily oriented 3d surfaces using a genetic algorithm. Pattern Recogn. Lett. **27**, 1201–1208 (2006)
51. Maintz, J., Viergever, M.A.: A survey of medical image registration. Medical Image Analysis **2**(1), 1 – 36 (1998)

52. 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)
53. Mian, A.S., Bennamoun, M., Owens, R.: Three-dimensional model-based object recognition and segmentation in cluttered scenes. IEEE Transaction Pattern Analisys Machine Intelligence **28**(10), 1584–1601 (2006)
54. Mian, A.S., Bennamoun, M., Owens, R.A.: Automatic correspondence for 3d modeling: an extensive review. International Journal of Shape Modeling **11**(2), 253–291 (2005)
55. 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)
56. 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)
57. Murino, V., Ronchetti, L., Castellani, U., Fusiello, A.: Reconstruction of complex environments by robust pre-aligned ICP. In: 3DIM (2001)
58. Myronenko, A., Song, X., Carreira-Perpinan, M.: Non-rigid point set registration: Coherent point drift. In: Neural Information Processing Systems Conference (2006)
59. 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-Verlag Berlin, Heidelberg (2008)
60. 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)
61. Park, K., Germann, M., Breitenstein, M.D., Pfister, H.: Fast and automatic object pose estimation for range images on the GPU . Machine Vision and Applications **21**(5), 749–766 (2009)
62. Park, S., Subbarao, M.: An accurate and fast point-to-plane registration technique. Pattern Recognition Letters **24**(16), 2967 – 2976 (2003)
63. Pears, N.E., Heseltine, T., Romero, M.: From 3d point clouds to pose normalised depth maps. International Journal on Computer Vision (2), 152–176 (2010)
64. 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)
65. Pissanetzky, S.: Sparse Matrix Technology. Academic Press (1984)
66. Pottmann, H., Huang, Q., Yang, Y., Hu, S.: Geometry and convergence analysis of algorithms for registration of 3D shapes. International Journal on Computer Vision **67**(3), 277–296 (2006)
67. Prasad, M., Zisserman, A., Fitzgibbon, A.W.: Single view reconstruction of curved surfaces. In: International Conference on Computer Vision and Pattern Recognition (2006)
68. Pulli, K.: Multiview registration for large data sets. In: 3DIM '99: Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling, pp. 160–168 (1999)
69. Pulli, K., Piiroinen, S., Duchamp, T., Stuetzle, W.: Projective surface matching of colored 3d scans. In: 3-D Digital Imaging and Modeling (3DIM), pp. 531–538 (2005)
70. Rangarajan, A., Chui, H., Duncan, J.: Rigid point feature registration using mutual information. Medical Image Analysis **3**, 425–440 (1999)
71. Rangarajan, A., H. Chui, E.M., Pappu, S., Davachi, L., Goldman-Rakic, P., Duncan., J.: A robust point-matching algorithm for autoradiograph alignment. Medical Image Analysis **1**, 379–398 (1997)
72. Ruiter, H.D., Benhabib, B.: On-line Modeling for Real-Time, Model-Based, 3D Pose Tracking. Springer (2007)
73. Rusinkiewicz, S., Brown, B., Kazhdan, M.: 3d scan matching and registration. ICCV Short Course (2005)
74. Rusinkiewicz, S., Hall-Holt, O., Levoy, M.: Real-time 3-D model acquisition. ACM Transactions on Graphics (Proc. SIGGRAPH) **21**(3), 438–446 (2002)

75. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: 3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on, pp. 145–152 (2001)
76. Salvi, J., Matabosch, C., Fofi, D., Forest, J.: A review of recent range image registration methods with accuracy evaluation. Image and Vision Computing **25**(5), 578–596 (2007)
77. Salzmann, M., Ilic, S., Fua, P.: Physically valid shape parameterization for monocular 3–D deformable surface tracking. In: British Machine Vision Conference (2005)
78. Sara, R.: Finding the largest unambiguous component of stereo matching. In: In Proc of European Conference on Computer Vision (ECCV), pp. 900–914 (2002)
79. Sara, R., Okatani, I., Sugimoto, A.: Globally convergent range image registration by graph kernel algorithm. In: 3-D Digital Imaging and Modeling (3DIM) (2005)
80. 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)
81. 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 Processing Mag. **27**(2), 50–60 (2010)
82. Sharp, G., Sang, L., Wehe, D.: Icp registration using invariant features. IEEE Trans. Pattern Anal. Mach. Intell. **24**(1), 90–102 (2002)
83. Silva, L., Bellon, O.R.P., Boyer, K.L.: Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **27**, 762–776 (2005)
84. Simon, D.A.: Fast and accurate shape-based registration. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA (1996)
85. Steinke, F., Scholkopf, B., Blanz, V.: Learning dense 3d correspondence. In: Annual Conference on Neural Information Processing Systems (NIPS 2006) (2007)
86. 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)
87. Tangelder, J., Veltkamp, R.: A survey of content based 3d shape retrieval methods. Multimedia Tools Application **39**(3), 441–471 (2008)
88. 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)
89. Trucco, M., Verri, A.: Introductory Techniques for 3-D Computer Vision. Prentice Hall (1998)
90. Tsin, Y., Kanade, T.: A correlation-based approach to robust point set registration. In: European Conference on Computer Vision, pp. 558–569 (2004)
91. Vinesh, R., Kiran, F.: Reverse Engineering, an industrial perspective. Springer (2008)
92. 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)
93. Watt, A.: 3D Computer Graphics. Addison-Wesley (2000)
94. 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)
95. Wyngaerd, J.V., Gool, L.V.: Automatic crude patch registration: Toward automatic 3d model building. Computer Vision and Image Understanding **87**(1-3), 8–26 (2002)
96. Zhang, Z.: Iterative point matching of free-form curves and surfaces. International Journal of Computer Vision (13), 119–152 (1994)
97. Zinsser, T., Schnidt, H., Niermann, J.: A refined icp algorithm for robust 3D correspondences estimation. In: International Conference on Image Processing, pp. 695–698 (2003)