

RESEARCH

Open Access

Multi-view alignment with database of features for an improved usage of high-end 3D scanners

Francesco Bonarrigo^{1,2}, Alberto Signoroni^{1,2*} and Riccardo Leonardi^{1,2}

Abstract

The usability of high-precision and high-resolution 3D scanners is of crucial importance due to the increasing demand of 3D data in both professional and general-purpose applications. Simplified, intuitive and rapid object modeling requires effective and automated alignment pipelines capable to trace back each independently acquired range image of the scanned object into a common reference system. To this end, we propose a reliable and fast feature-based multiple-view alignment pipeline that allows interactive registration of multiple views according to an unchained acquisition procedure. A robust alignment of each new view is estimated with respect to the previously aligned data through fast extraction, representation and matching of feature points detected in overlapping areas from different views. The proposed pipeline guarantees a highly reliable alignment of dense range image datasets on a variety of objects in few seconds per million of points.

Keywords: 3D modeling pipelines, Geometric feature extraction, Feature description and similarity, Partial shape matching, Range image datasets, 3D scanner usability

Introduction

In the last years, 3D datasets acquired with modern optical scanning devices (either based on structured light projection or laser beams) increased their spatial resolution. Other primary features, such as accuracy and acquisition speed, have also been improved. Scanner are decreasing their size and weight and their usage is expected to become more and more intuitive and unconstrained, such as the use of digital cameras. This would be desirable in response to an increasing demand of “3D” for today professional applications (industry and design, medicine, cultural heritage, robotics, mechanics, building constructions...) as well as for soon to come web applications.

The first step toward the 3D modeling of a physical object is the acquisition of multiple scans from different viewpoints of the surface of interest. After the acquisition, each set of 3D data (either composed by range images or point clouds) needs to be accurately aligned (or registered)

into a common coordinate system [1-3]. Aligned datasets are then fed to a subsequent step of the object modeling pipeline [4], with the initial alignment quality influencing the accuracy and fidelity of the resulting object model.

Under the basic hypothesis of a certain degree of overlap among different views, the registration problem can be conceptually split into a cascade of *coarse* and *fine* alignments. Even if they can be considered as instances of a more general class of problems called ‘shape correspondence’ [5], they are different in nature and require distinct solving approaches. The goal of coarse alignment is to roto-translate each independently positioned scan (an example is shown in Figure 1) so that the entire dataset can be roughly brought into a common reference system (as shown in Figure 2). This takes place without making any assumption on the initial viewpoints. A second stage fine alignment is then used to accurately register the scans from the first approximate alignment.

Both coarse and fine approaches can be performed *pair-wise* or *multi-view*. Solutions within the first class only exploit the information present in the overlapping area with respect to a single range image to determine separately the alignment between each pair of views, while

*Correspondence: alberto.signoroni@ing.unibs.it

¹Information Engineering Dept., University of Brescia, Brescia, Italy

²CNIT - UoR at University of Brescia, Brescia, Italy

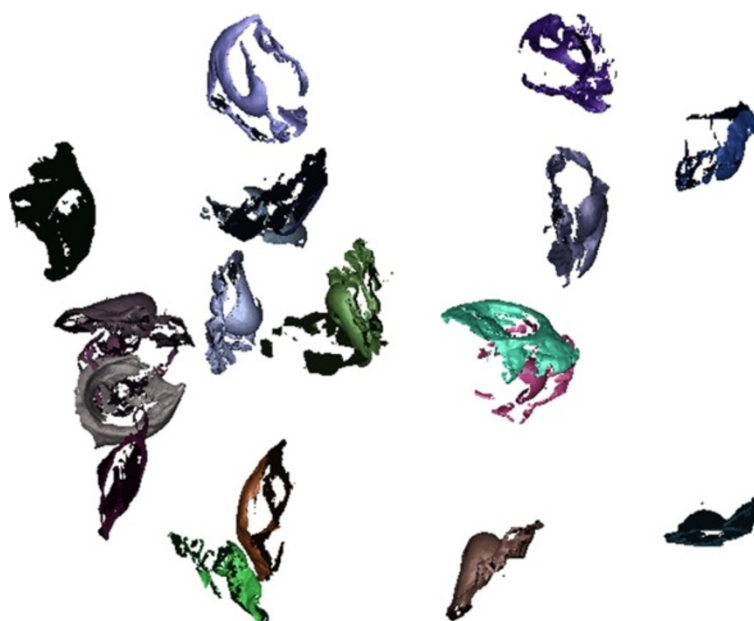


Figure 1 A set of scrambled views composing the Dolphin object prior coarse alignment application.

multi-view alignment integrates all the available overlapping information collected from all acquired scans and exploits it in order to estimate the correct alignment. Even though pairwise and multi-view coarse alignment approaches may seem equally effective to accomplish their role, this is not true anymore in a practical acquisition settings. In fact, pairwise coarse alignment requires each couple of subsequent views in a dataset to possess a certain amount of overlap (from now on we address such requirement as the *overlap constraint*). This can be obtained by either forcing the operator to devise in advance an acquisition path which complies with the overlap constrain (thus limiting the scanner usability),

or by manually reordering the original acquisition path as a preprocessing step (which may represent a tedious and time consuming process). On the contrary, multi-view approaches allow a more unconstrained usage of the acquisition devices, in that the overlap condition can be relaxed so that each incoming scan is only required to possess a certain overlap with respect to *any* of the previously aligned range data. This effectively extends the applicability of automated coarse alignment to a greater number of less constrained acquisition paths. From now on, we shall address as *unchained path* any acquisition path which abides by the previously stated relaxed requirement.

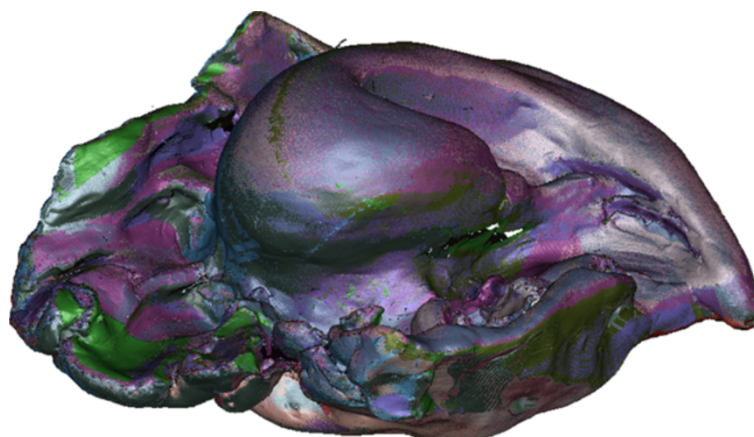


Figure 2 The realigned Dolphin after application of the coarse alignment.

In this work, our main focus is on the description of a new multi-view coarse alignment method suited for high-resolution datasets, specifically designed to fulfill the following requirements: (1) it should improve the scanner usability by allowing the operator the freedom to choose the acquisition path, (2) it should prove to be equally effective regardless to the nature of the acquired object as well as its size, (3) the alignments should be fast so as to give the user on-the-fly visual feedbacks during object digitalization. We will see that effective multi-view alignment solutions can be designed starting from an efficient pairwise alignment technique. In this paper we present the proposed solutions in two main parts: pairwise alignment (Section “Pairwise alignment technique”) and multi-view alignment (Section “Multi-view alignment pipeline”). In particular we propose:

- for Pairwise alignment:
 - (1) a multiscale feature extraction technique capable of identifying feature points and their scale (Section “Feature extraction”);
 - (2) a lightweight feature signature devised to quickly reduce the matches space (Section “Feature description”);
 - (3) a matching chain developed to progressively skim the correspondence space (Sections “Feature matching” and “Correspondence test and selection”);
- for Multi-view alignment:
 - (1) the definition of a feature database for the multi-view alignment (Section “Feature database update”);
 - (2) the use of a global adjustment solution to optimize the alignment obtained through the proposed multi-view pipeline (Section “Global adjustment”);

Each block of the pipeline has been tested in isolation (Section “Comparative tests”). An evaluation of the entire pipeline performance is proposed (Section “Experimental results”) on a set of range image datasets, presented in Section “Acquired datasets”, representing different objects taken during real acquisition campaigns. In particular, in-depth comparative tests with respect to alternative solutions have been made both on the feature extraction and description phases (Section “Feature extractor and descriptor comparison”) and on the correspondence test and selection solutions (Section “Correspondence test technique comparison”). Overall performance evaluations are presented in (Section “Experimental results”) and organized accordingly to the structure of the proposed solutions: Section “Pairwise alignment results” presents a quantita-

tive evaluation of the pairwise alignment performance in terms of success rate and computational complexity, while Section “Multi-view alignment pipeline” presents a quantitative evaluation of the multiview alignment in terms of success rate, computational complexity and alignment error. Conclusion are drawn in Section “Conclusions”. The present paper builds on our conference paper [6], extending it in several parts. The experimental section has been widened considerably, while performance evaluation of each pipeline block in isolation has been performed.

Related work

Multiple scan alignment without prior knowledge of the scanning viewpoints is a classic problem in 3D modeling which have found several solutions in the computer vision literature [7]. Some representative works are [1-3,7-9]. Being the first step of a modeling chain, its performance is of crucial importance for the attainment of the final object model, since a certain degree of accuracy is strongly required by the subsequent fine registration steps to converge to the correct solution. In fact it is well known that, for fine alignment, classic solutions (e.g. ICP [10] and its variants [11]) are based on optimization routines which often suffer from convergence to local minima which should be maximally reduced by proper initialization of each view referencing. Coarse alignment solutions can be traced back to one of the two main philosophies that have emerged during these years, i.e. with or without the exploitation of feature descriptors. The first approach exploits the ever-increasing computational capabilities of modern calculators to find, within a large solution space, the affine transform that better aligns two views. The main advantage of the techniques which fall into this category is that they are independent from the input data and are more robust to noisy data. On the other hand, they are usually computationally expensive. The progenitor of this family is considered to be the RANSAC, devised by Fischler and Bolles [12]. During the years, improvements to this algorithm have been proposed in order to reduce the computation time, also by exploiting point neighborhood descriptors [13,14]. A second approach for coarse registration relied on the extraction and subsequent matching of (global, local or multiscale) shape descriptors [7,15]. Advantages with respect to brute-force approaches are mainly related to the computational gain achieved through an accurate selection and skim of descriptive features. On the other hand, they usually fail in describing featureless surfaces, and are more sensitive to noise. Feature-based approaches are widely used in several application fields such as similarity search, object retrieval and categorization [16-18], shape correspondence and analysis [5,15]. Multiscale feature-based approaches (such as the one we propose in this work) allow a better adaptation to features of different

kind and dimension. Related works are those presented by Li and Guskov [19] and Lee et al. [20] which introduced extensions of Lowe's 2D SIFT [21] to 3D datasets. Their approach has been subsequently exploited by Castellani et al. [22] and Bonarrigo et al. [6]. SIFT-related feature description distinctiveness against Spin Images has been compared in [23]. Thomas and Sugimoto [24] proposed to use the reflectance properties for images registration to better perform on featureless images. In feature-based approaches, feature points are usually associated to descriptors, which ideally should associate an unambiguous signature to each feature, which is fast to compute and robust to any viewpoint rotation as well as to variations of point density on the image. Li and Guskov [19] proposed a descriptor based on a combination of local Discrete Fourier transform and Discrete Cosine transform to describe the neighborhood of each feature point. Gelfand et al. in [3] proposed the use of volumetric descriptors, that is the estimation of the volume portion inscribed by a sphere centered at a point of the surface. Castellani et al. [22] proposed a statistical descriptor based on a hidden Markov chain that is trained through its neighborhood. Despite scan alignment and similar problems have been thoroughly explored in the literature, meaningful comparisons remain somehow difficult to accomplish for several reasons: the heterogeneity of each approach, the different way in which they are combined into functional alignment pipelines, the variability of the addressed performance and application requirements, the differences of test datasets both in terms of representation primitives (range images, point clouds, meshes, etc.) and data characteristics (resolution, noise, etc.) related to the differences in data acquisition setup and scanning hardware. In particular, even if each block of a feature-based alignment pipeline (feature extraction, description, matching, correspondence selection and transform estimation) can be tested in isolation with respect to alternative solutions (we will do this in Section "Comparative tests"), critical interdependencies exist among the different pipeline blocks. In this work we adopted design criteria which take full account of such interdependency, according to what has been recognized in reference reviewing work [15,17]. Bustos et al. [17] indicated that a complete and fair comparison of feature extraction and description techniques is unfeasible and that the specific application requirements should inspire and provide guidance for the design of application-driven solutions. Bornstein et al. [15] concluded their review by suggesting to avoid the design of pipelines composed by unnatural block combinations. Another relevant aspect is that multiple-view coarse alignment solutions are usually proposed in literature as simple extensions of pairwise approaches. This leads to a serious limitation on the usability of the acquisition devices. The increased complexity of a multiple-view

coarse alignment has been recognized in [1] with a solution proposed by model graphs and visibility consistency tests. An approach based on view trees has been adopted in [9] where multiple-view coarse alignment is obtained by maximization of inlier point pairs. However, these solutions do not allow interactive acquisition because they work on a graph of the pairwise matches once all the alignments are completed, and this conflicts with the need of on-the-fly visual feedbacks. By exploiting the possibility of some range scanners to acquire views at high frame rate (e.g. Kinect and alike sensors), hand-held on-the-fly acquisition and alignment solutions have been proposed [25,26] with specific solutions for handling loop closure problems [27] or computational burden [28]. However, these interesting applications are still too far from today professional requirements of high point density and metric precision.

Pairwise alignment technique

Overview and notation

In this section we shall introduce some notation, as well as briefly summarize the alignment process. A range image can be conceived as the projection of a 2D image grid on a 3D target object surface and the acquisition of depth-related information from that surface. The resulting dataset is a "structured" point cloud, that is a set of points lying in a 3D space, and associated to a pixel of the acquisition grid. We define a range image as a map $I \subset \mathbb{Z}^2 \rightarrow R \subset \mathbb{R}^3$, where the domain I is a rectangular grid (usually corresponding to the CCD matrix), while the co-domain R corresponds to the set of 3D points representing the acquired surface. Due to the limitations of the acquisition procedure (limited measure range, occlusions due to the object shape, etc.), not all pixel positions $i \in I$ may have a valid corresponding point $\mathbf{p}_i \in R$, therefore only a subset $I_V \subseteq I$ of valid points is acquired for each image. We take advantage of range images data structure in order to speed up the processing: in particular, by exploiting the image domain I , neighborhood information can be retrieved quickly and efficiently, while data processing is performed over the 3D target space R . An example of efficient processing obtained by exploitation of the 2D grid is the normal field estimation method we employ. For a given point \mathbf{c} we exploit the 2D grid to identify its 8 closest neighbours \mathbf{p}_k , with $k \in [0, 7]$, to estimate the normal $\hat{\mathbf{n}}_{\mathbf{c}}$. At first, for every valid point \mathbf{p}_k the vectors $\mathbf{v}_k = \mathbf{p}_k - \mathbf{c}$ are computed. Then the outer products between each valid couple of vectors ($\mathbf{v}_k, \mathbf{v}_{(k+2)\%8}$) are determined. Once normalized, they are summed together and their average is set as the normal for point \mathbf{c} .

Figure 3 describes the pipeline we adopt for the pairwise alignment between the pair of range images RI_a and RI_b . All the building blocks highlighted in red are recalled in subsequent sections, while in the following we briefly summarize the alignment process. Given a

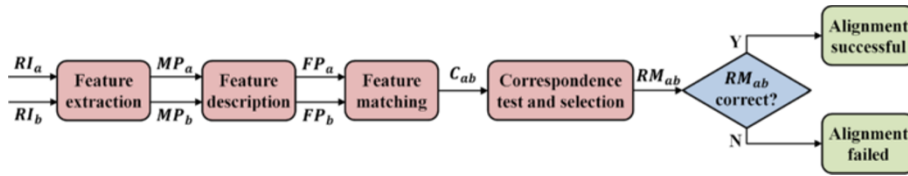


Figure 3 The pairwise alignment system, composed by: feature extraction (Section “Feature extraction”), feature description (Section “Feature description”), feature matching (Section “Feature matching”), correspondence test and selection (Section “Correspondence test and selection”).

pair of range images RI_a and RI_b , the first step in the pipeline applies a multi-scale feature extraction method thoroughly described in Section “Feature extraction”, such that meaningful points MP_a and MP_b are identified on the respective surfaces. Following, each point in the sets MP_a and MP_b is associated with a signature, described in Section “Feature description”, which encodes neighborhood information with respect to the meaningful point. The signature is designed to be suitably discriminative and fast to compute. This way, two sets of feature points FP_a and FP_b are obtained. Correspondences between these 2 sets are then established through the matching process described in Section “ssec:featureMatching”, which exploits the signatures in order to identify pairs of compatible feature points so that a correspondence set C_{ab} is determined. Each correspondence c is constituted of a pair of feature points, one taken from FP_a and the other from FP_b . In the next phase, correspondences are grouped to form triplets, as described in Section “Correspondence test and selection”, and the ones which are most likely to be correct are collected in a triplet set T_{ab} . We resort to triplets of correspondences since they contain the minimum number of points (that is, 6 3D coordinates, 3 of which lie in the reference system associated to RI_a while the others lie in the one associated with RI_b) for which a roto-translation matrix can be estimated, for instance applying the method described by Horn in [29]. The triplet set T_{ab} is then evaluated so that a single roto-translation matrix RM_{ab} is identified and verified with respect to its correctness.

Feature extraction

The feature extraction technique we adopt here is a modified version of the approach proposed by Castellani et al. [22]. His approach is similar to the one introduced by Lee et al. [20], which can be in turn considered as a 3D extension of the 2D multi-scale analysis proposed by Lowe [21]. Hereafter we shall summarize Lee’s approach, then we will highlight the modifications introduced by Castellani, as well as our own.

The approach from Lee et al. requires that: (a) given a range image RI , M filtered images $G(r)$, at scales $r \in [1, M]$, are derived by applying Gaussian kernels of

growing dimension; (b) a set of $M - 1$ saliency maps $S(r)$ is derived from pairs of $G(r)$ at consecutive scales, from which a set of MP is identified. These MP are candidate features, and are associated with the scale at which they have been detected. To produce the filtered images $G(r)$ at various scales $r \in [1, M]$, a geometric Gaussian filtering is applied to each valid point p_i in the RI, obtaining $g_i(r)$:

$$g_i(r) = \frac{\sum_{p_j \in B_{2\sigma_r}(p_i)} p_j \cdot e^{-\frac{\|p_i - p_j\|^2}{2\sigma_r^2}}}{\sum_{p_j \in B_{2\sigma_r}(p_i)} e^{-\frac{\|p_i - p_j\|^2}{2\sigma_r^2}}} \quad (1)$$

where $B_{2\sigma_r}(p_i)$ identifies the points within a euclidean distance $2\sigma_r$ from p_i . A filtered image $G(r)$ is thus defined as the set of points $g_i(r)$, with $i \in [1, |I_V|]$. As the kernel radius σ_r increases, details which size is smaller than σ_r are smoothed out from $G(r)$ and, when the kernel size doubles, computations are performed on images subsampled by a factor two.

Once the filtered versions $G(r)$ have been calculated, $M - 1$ saliency maps are derived. A saliency map is a 2D array of scalar values, obtained by pairwise subtraction of $G(r)$ at adjacent scales. This retains only the details comprised between the two bounding scales r and $r + 1$, in other words it highlights features which dimension is comprised between the kernel sizes σ_r and σ_{r+1} . Saliency maps $S(r) = \{s_i(r)\}$ are calculated as follows:

$$s_i(r) = \|g_i(r) - g_i(r + 1)\| \quad (2)$$

The maximum values of each saliency map $S(r)$ are considered to be MP and located through an iterative search where, once the greatest valid saliency value for $S(r)$ is found, no other maximum can be selected within an invalidation neighborhood region $B_{2\sigma_{r+1}}(p_i)$. This prevents the selection of feature points which descriptors would partially overlap, since the descriptor size for features detected at scale $S(r)$ is σ_{r+1} .

With respect to Lee’s approach, Castellani et al. proposed that, for each valid 3D point, its saliency is estimated by calculating the projection norm between the

pairwise difference and the normal associated to the original 3D point:

$$s_i(r) = \|\langle \hat{n}_i, \mathbf{g}_i(r) - \mathbf{g}_i(r+1) \rangle\| \quad (3)$$

This correction should help in reducing the saliency values associated to points that, after the filtering, have moved away from their normal direction. We have found out, however, that such saliency correction (originally proposed for mesh data) needs to be corrected in order to apply it to range images and point clouds. Since these types of data usually contain a greater number of 3D points than mesh vertices, their normal field is more subject to vary due to the filtering process. Therefore, in order to obtain a reliable saliency estimation, we propose to use the normal field associated to the filtered data itself:

$$s_i(r) = \|\langle \hat{n}_i(r), \mathbf{g}_i(r) - \mathbf{g}_i(r+1) \rangle\| \quad (4)$$

Although such a modification requires an estimation of the normal field for every filtered image $G(r)$, we will demonstrate in Section “Feature extractors comparison” that it brings a significant performance boost with respect to feature localization.

This saliency estimation described in (4), brought a slight increment in performance with respect to the approach initially proposed in [6]. Once the set of MP has been computed, a number of tests on each meaningful point is performed in order to make sure that: (1) its neighbour points are well distributed and (2) it is not close to a border or hole, otherwise the associated descriptor would be incomplete; (3) it does not lie over a saliency ridge, because in such cases small variations in saliency

estimation may cause great variations of maximum localization. If any of these conditions is not met, the point is discarded from the set MP. Hereafter, for a neater and more compact notation we will omit unnecessary indices when things have general validity.

Feature description

In order to search for correspondences between feature points belonging to different range images we rely on feature descriptors. We propose to associate to each feature point $\mathbf{f} \in \text{MP}$ detected at scale σ_r a descriptor which encodes information extrapolated from both the normal vectors and saliency data of the neighbour points $\mathbf{p}_j \in B_{\sigma_{r+1}}(\mathbf{f})$. In summary, given a feature point \mathbf{f} , we define a polar grid spanning the tangent plane to \mathbf{f} . This grid is composed of M radial sectors and L angular sectors, as shown in Figure 4. Every neighbour point \mathbf{p}_j within $B_{\sigma_{r+1}}(\mathbf{f})$ is associated to a grid sector, and for each sector we compute two score values derived by the normal and saliency information. As a result, our descriptor is composed by $2 \times M \times L$ real (floating point) values. Following, we describe in more detail how the descriptor is computed.

At first, a local reference system $\hat{x}_f, \hat{y}_f, \hat{z}_f$ is centered on \mathbf{f} . Versor \hat{z}_f is set toward the direction of the feature normal \hat{n}_f , while the span of $\{\hat{x}_f, \hat{y}_f\}$ identifies \mathbf{P}_f , i.e. the plane tangent to \mathbf{f} . The direction of \hat{x}_f is randomly selected, while $\hat{y}_f = \hat{z}_f \times \hat{x}_f$. A polar grid of radius σ_{r+1} is then defined and subdivided into M radial and L angular sectors. We have empirically found that $M = 3$ and $L = 36$ generate a distinctive signature, while allowing fast computation. Each point \mathbf{p}_j belonging to $B_{\sigma_{r+1}}(\mathbf{f})$ is then associated to a given sector of the polar grid by mapping \mathbf{p}_j on the plane \mathbf{P}_f . This is represented in Figure 4 by the

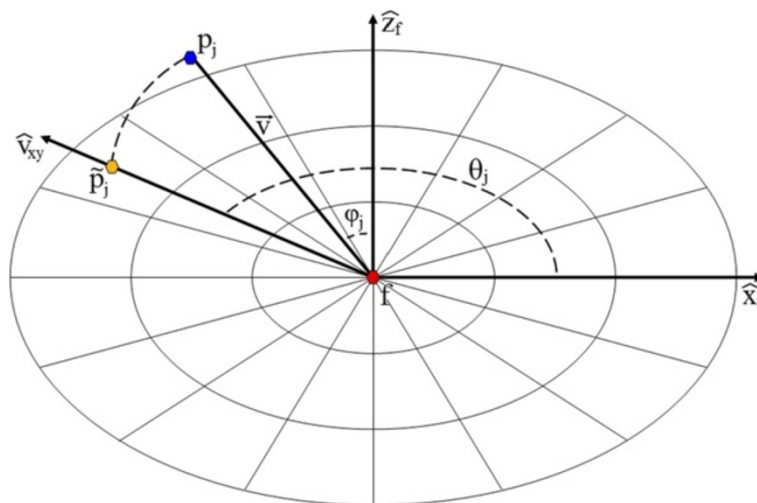


Figure 4 Signature grid description.

point $\tilde{\mathbf{p}}_j = \mathbf{f} + \|\mathbf{v}\| \cdot \hat{\mathbf{v}}_{xy}$, where $\mathbf{v} = \mathbf{p}_j - \mathbf{f}$. The sector indexes (m_j, l_j) associated to \mathbf{p}_j are computed as follows:

$$m_j = \left\lfloor \|\mathbf{v}\| \frac{M}{\sigma_{i+1}} + 0.5 \right\rfloor \quad l_j = \left\lfloor \theta_j \frac{L}{2\pi} + 0.5 \right\rfloor \quad (5)$$

where

$$\begin{aligned} \theta_j &= \begin{cases} \arccos(\langle \hat{\mathbf{v}}_{xy}, \hat{\mathbf{x}}_f \rangle) & \langle \hat{\mathbf{v}}_{xy}, \hat{\mathbf{y}}_f \rangle \geq 0 \\ 2\pi - \arccos(\langle \hat{\mathbf{v}}_{xy}, \hat{\mathbf{x}}_f \rangle) & \langle \hat{\mathbf{v}}_{xy}, \hat{\mathbf{y}}_f \rangle < 0 \end{cases} \\ \varphi_j &= \arccos(\langle \hat{\mathbf{v}}, \hat{\mathbf{z}}_f \rangle) \\ \hat{\mathbf{v}}_{xy} &= \frac{\mathbf{v} - \|\mathbf{v}\| \cos(\varphi_j) \cdot \hat{\mathbf{z}}_f}{\|\mathbf{v} - \|\mathbf{v}\| \cos(\varphi_j) \cdot \hat{\mathbf{z}}_f\|} \end{aligned} \quad (6)$$

Once each point $\mathbf{p}_j \in B_{\sigma_{r+1}}(\mathbf{f})$ has been associated to a sector, it is possible to compute \mathbf{w}_f , the descriptor associated to feature point \mathbf{f} . At first, for each sector (m, l) the average normal vector $\hat{\mathbf{n}}(m, l)$ and saliency $s(m, l)$ are computed (if a sector does not contain any point, it is considered invalid). Then, given $\hat{\mathbf{n}}_f$ and s_f respectively the normal vector and the saliency value associated to the feature point \mathbf{f} , the sector descriptor $\mathbf{w}_f(m, l)$ is computed as follows:

$$\begin{aligned} \mathbf{w}_f(m, l) &= [\Delta n(m, l), \Delta s(m, l)] \\ \Delta n(m, l) &= 1.0 - |\langle \hat{\mathbf{n}}(m, l), \hat{\mathbf{n}}_f \rangle| \\ \Delta s(m, l) &= 1.0 - \frac{s(m, l)}{s_f} \end{aligned} \quad (7)$$

Each sector descriptor $\mathbf{w}_f(m, l)$ will therefore contain two scores (Δn and Δs , both $\in [0, 1]$) related to how much the average normal and saliency values in each grid sector differ from the corresponding feature point values. The feature descriptor \mathbf{w}_f is therefore the set of the $M \times L$ sector descriptors $\mathbf{w}_f(m, l)$ thus calculated. Once each feature point has been associated to its descriptor, they are organized in the set FP for the subsequent processing stages.

The proposed descriptor is fast to compute since both normals and saliency information are already available (the latter being a byproduct of the extraction procedure described in Section “Feature extraction”) once the feature points have been identified. Moreover, it is moderately lightweight as it only requires $2 \times M \times L$ floating point values. Nevertheless, it is enough selective as to allow to skim the correspondence space to a more treatable dimension while retaining enough correct correspondences. In Section “Feature descriptor comparison” we will complete these observations by comparing our descriptor to the Spin Images introduced by Johnson [30] and by demonstrating its superior performance.

Feature matching

Given two feature sets FP_a and FP_b , estimated on different range images, the objective of this pipeline block is to

calculate all possible feature correspondences by matching the descriptors of each pair of features ($\mathbf{f}_s \in \text{FP}_a$, $\mathbf{f}_d \in \text{FP}_b$), and subsequently select a subset of correspondences C_{ab} which is likely to contain the correct ones. Figure 5 gives the reader an intuition on how feature similarities can define relevant matches.

In Section “Feature description” we stated that the direction of $\hat{\mathbf{x}}_f$ is chosen arbitrarily for each feature descriptor. This can be modeled with an unknown displacement factor \bar{L} , with $\bar{L} \in [1, L]$, between the signatures of the same feature point on two different range images. We therefore compensate this indetermination by performing an operation similar to the circular correlation, in that one of the descriptors remain fixed, while the other is shifted along the circular direction for L times. At each rotation step a score is computed, and the maximum score value is selected as follows:

$$c_{sd}^{\text{score}} = \max_{\bar{L} \in [1, L]} \{c_{sd}^{\text{score}}(\bar{L})\} \quad (8)$$

with

$$\begin{aligned} c_{sd}^{\text{score}}(\bar{L}) &= \sum_{m=1}^M \sum_{l=1}^L [n_{sd}^{\text{score}}(m, l, \bar{L}) \cdot s_{sd}^{\text{score}}(m, l, \bar{L})] \\ n_{sd}^{\text{score}}(m, l, \bar{L}) &= (1 - |\Delta n_s(m, l) - \Delta n_d(m, \bar{L})|) \\ s_{sd}^{\text{score}}(m, l, \bar{L}) &= (1 - |\Delta s_s(m, l) - \Delta s_d(m, \bar{L})|) \end{aligned}$$

Once all possible matches between the two feature sets FP_a and FP_b have been computed, we need to skim the matches set from its original size $|\text{FP}_a| \cdot |\text{FP}_b|$ to a more treatable dimension. We therefore define a correspondence set C_{ab} of size Q as the list of correspondences c_q found between FP_a and FP_b which possess the highest correspondence score. We decided to retain the best Q correspondences (in this work we use $Q = 150$), rather than fix a hard threshold for the score, since the score distribution is not constant with respect to different image pairs. Within the set of Q best matches we have also verified that, in general, the majority of correct correspondences share the top positions of the ranking together with a small number of false matches (generated by incidental signature similarities). This means that we cannot “blindly” exploit the first correspondences of the ranking to estimate the roto-translation matrix between the views. We therefore introduce a robust selection step (described in the following section) to determine the most reliable correspondences within the set C_{ab} .

Correspondence test and selection

In order to determine a roto-translation matrix that references RI_b to RI_a , at least 3 correct correspondences

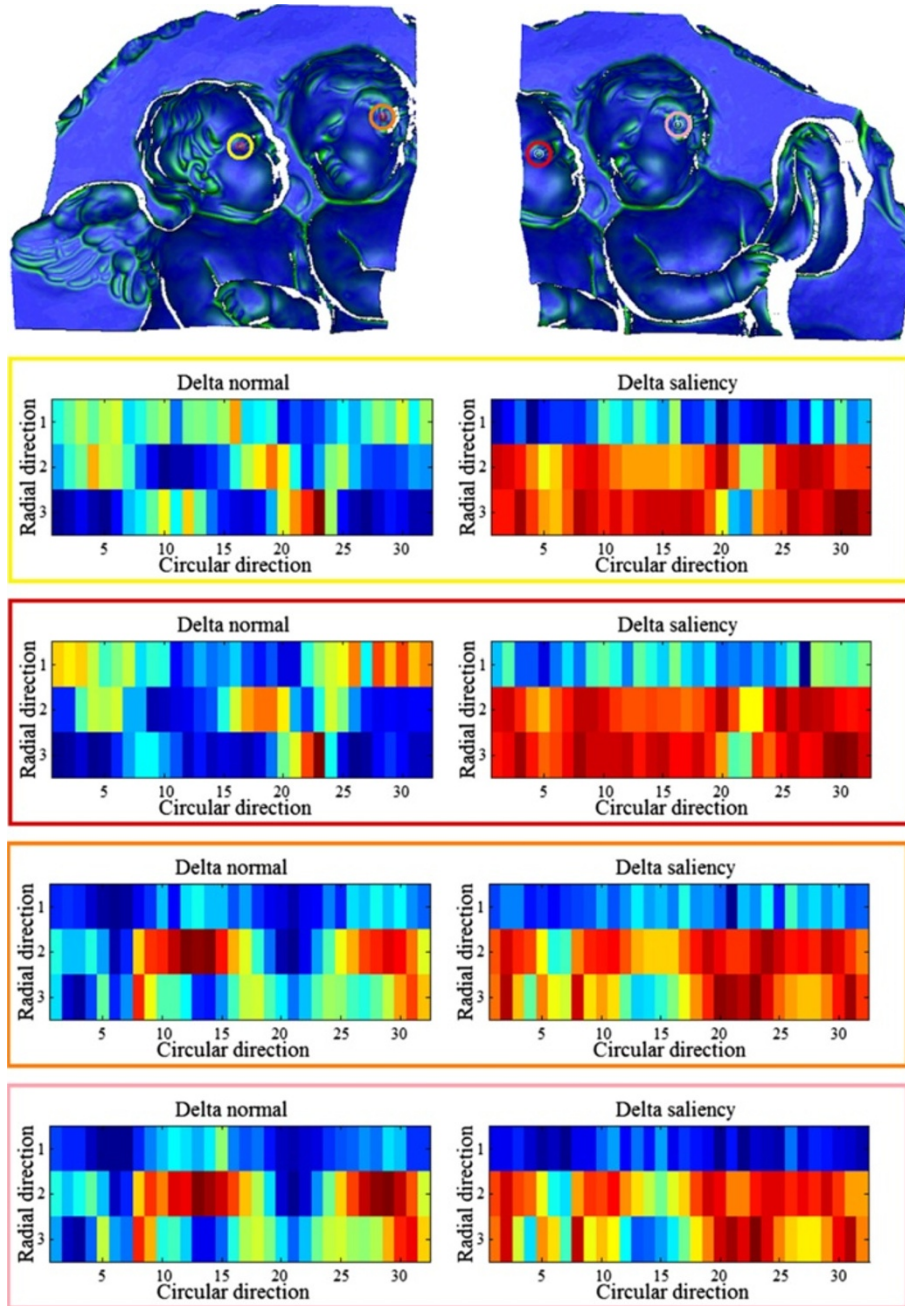


Figure 5 Feature signatures. Upper part: two range images on which some feature points are highlighted with different colors. Below, graphical visualization in a red-blue scale of the signatures, contoured with their corresponding colors. The first two features (yellow and red) as well as the last two ones (orange and pink) have well matching descriptors.

(a triplet) need to be identified within the set C_{ab} . Each triplet t is defined as follows:

$$t = \{c_g, c_h, c_j\}, \quad \text{with} \quad \begin{cases} c_g, c_h, c_j \in C_{ab} \\ g, h, j \in [1, Q] \\ g \neq h \neq j \end{cases} \quad (9)$$

Given the correspondence set C_{ab} of size Q , the number of possible non-repeating triplets is $(Q^3 - 3Q^2 + 2Q)/6$. Determining which (if any) of the triplets is correct is a computationally expensive task: for Q equal to 150 we would obtain more than half million triplets, therefore brute-force approaches such as directly test each of the possible roto-translations is not a viable option.

Solutions such as RANSAC [12] and similar ones are known to be effective solutions in performing these type of searches, however it is also known that the number of trials required to determine the correct model (in our case, a triplet) grows exponentially with respect to the proportion of outliers in the set. Unfortunately, in our scenario the number of inliers within the set C_{ab} is likely to be small. Indeed, it is below 15% in more than half of the considered cases, as we will show in Section “Correspondence test technique comparison”. Therefore we can expect the computational burden of RANSAC-style approaches to be high for such scenario.

Alternatively, we propose a robust selection procedure designed to progressively skim the correspondences so that the computational cost is kept low at each stage. The procedure consists of three steps: (1) every correspondence within C_{ab} is validated against each other and a score is calculated for each pair of matches; (2) for each triplet of correspondences a score is computed based on the three pairwise scores previously calculated, and a subset T_{ab} of U best triplets is retained; (3) for each triplet in T_{ab} , a roto-translation matrix RM is estimated and applied to the feature set FP_b , corresponding points are searched within image RI_a . The triplet which collects the highest number of such corresponding points is considered as the more reliable estimate. The above three steps are now described in detail.

(1) In order to validate each correspondence with respect to the others we exploit the rigidity constraint, which states that the distance between two points subject to an Euclidean transformation remains constant. We also introduce the concept of relative distance between a pair of correspondences, illustrated in Figure 6, and defined as follows:

$$d_{gh} \equiv d(c_g, c_h) = \frac{\|p_g^A - p_h^A\| - \|p_g^B - p_h^B\|}{\max(\|p_g^A - p_h^A\|, \|p_g^B - p_h^B\|)} \quad (10)$$

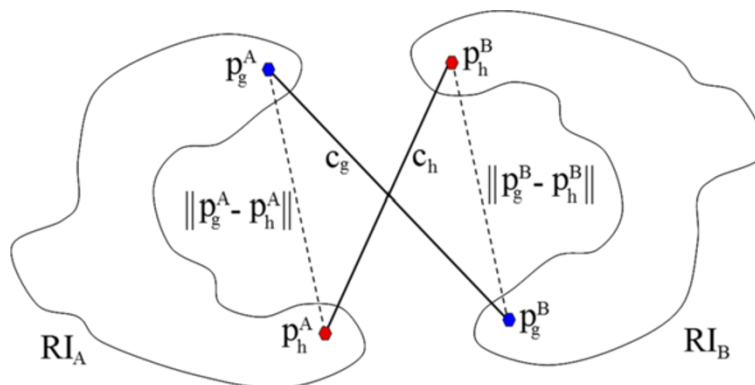


Figure 6 Example of correspondences distance d_{gh} of (10).

Due to the normalization term at the denominator, the relative distance is bounded between 0 (same distance) and 1 (maximum distance). The evaluation of the relative errors allows to perform a more unbiased ranking than that it would be with absolute errors (which tend to favor correspondence pairs of features which are close together). Once the relative distances have been estimated, they are organized into a $Q \times Q$ matrix **DM**:

$$\mathbf{DM} = \begin{bmatrix} 0 & d_{12} & d_{13} & \cdots & d_{1Q} \\ d_{21} & 0 & d_{23} & & d_{2Q} \\ d_{31} & d_{32} & 0 & & d_{3Q} \\ \vdots & & & \ddots & \vdots \\ d_{Q1} & d_{Q2} & d_{Q3} & \cdots & 0 \end{bmatrix} \quad (11)$$

DM is symmetric ($d_{hg} = d_{gh}$), and possesses zeros over its main diagonal ($d_{gg} = 0, \forall g \in [1, Q]$). An example of how such matrix looks like is presented in Figure 7.

(2) Once **DM** is calculated, the triplet space is skimmed by determining the set T_{ab} of U triplets (we set U to 25) which present the maximum value of the following score:

$$t^{\text{score}} = 1 - \frac{d_{gh} + d_{hj} + d_{jg}}{3} \quad \begin{cases} g, h, j \in [1, Q] \\ g \neq h \neq j \end{cases} \quad (12)$$

Similarly to the skim procedure described in the previous section, this triplet skim procedure is able to retain the triplets made of correct matches in the highest positions of the ranking.

(3) In order to determine the most correct triplet within the set T_{ab} , for each $t_u \in T_{ab}, u \in [1, U]$ the following steps are performed:

- the roto-translation matrix RM_u associated to triplet t_u is estimated through Horn method [29];
- the feature set FP_b is roto-translated through application of RM_u ;
- corresponding points between FP_b and RI_a are identified.

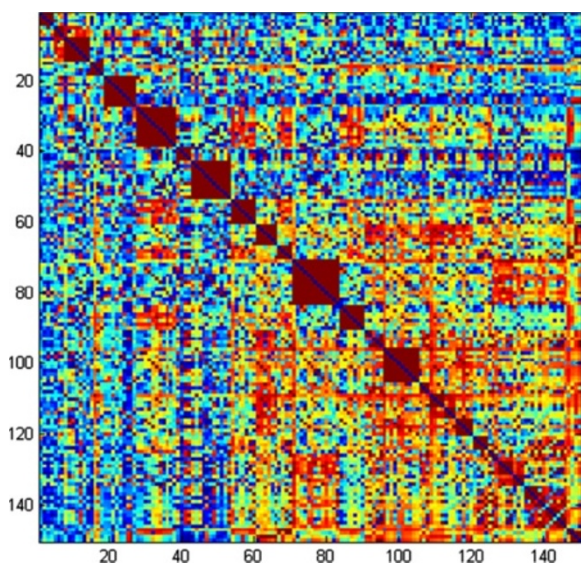


Figure 7 A distance matrix DM: blue dots represent low relative distance, while red ones identify distant matches. The red square clusters present along the diagonal are generated whenever evaluating pairs of correspondences that share one feature point (leading to a relative distance of 1).

The triplet \bar{t} which is found to possess the higher number of corresponding points is considered as the one that is most likely to be correct. Its associated roto-translation matrix $\bar{R}\bar{M}$ is thus refined by taking into account all the corresponding points just estimated. At last, the obtained alignment is tested by selecting a subset of points from RI_b , roto-translating them through $\bar{R}\bar{M}$, and verifying that at least a given percentage of points find a correspondence in RI_a . If the number of matches is above this threshold, image RI_k is considered as successfully aligned to the previous one. Coherently with the view overlap constraint we set that threshold to 20%. As we will demonstrate in Section “Correspondence test technique comparison”, given the typical proportion of inliers for the set C_{ab} , the speed of our approach outperforms both RANSAC [12] as well as its descendant PROSAC [31] in determining the best triplet.

Multi-view alignment pipeline

As we will see in the experimental sections, the described pairwise alignment pipeline, even compared to other solutions, presents good performance and is suitable for automatic alignment of dense range scans in terms of computational speed and robustness (successful alignment rate). Nevertheless, in order to reconstruct an entire object through such a pipeline, we would need to either define a chained acquisition path for which every consecutive image pair complies with the overlap constraint

(thus limiting the scanner usability), or manually reorder the scans prior to applying the alignment (a tedious, time consuming and necessarily off-line operation). Therefore, from both a conceptual and practical point of view the chained path assumption turns out to be a limitation to the scanner usability.

For instance, imagine that an operator is using a scanner to acquire a human statue, trying to abide by the policy imposed by the overlap constraint. He starts acquiring the front of the statue (chest, belly, etc.), then progressively moves toward the back. At a certain point, however, he realizes that some region on the belly was not acquired properly, creating a “hole” in the digital model. He therefore decides to acquire another range image on the front so that the hole can be properly filled. However, in order to comply with the overlap constraint he would need to capture more range data in regions he has already scanned to create a path between the back and the belly of the statue. It would be much simpler if he could follow an *unchained* acquisition path, as defined in Section “Introduction”. Another advantage of an unchained path multi-view alignment with respect to the pairwise approach is the fact that overlapping regions from previously aligned views can be cumulated to improve the chances of alignment: one view may not present enough overlap with respect to all the previously acquired ones, in such case any pairwise alignment attempt is likely to fail due to lack of reliable correspondences or because the 20% overlap requirement stated in Section “Correspondence test and selection” is not met although the union of overlapping regions taken from different views may contain enough overlap (as well as reliable correspondences) to achieve the alignment.

We then extend the pairwise approach by working with a database of features (indicated with FP_{db}) which collects the feature sets associated to all the previously aligned range images, so that the feature set associated to the current range image FP_k is matched to the database FP_{db} . Figure 8 illustrates the block diagram of the multi-view alignment pipeline we propose. With respect to the stages described in Section “Pairwise alignment technique”, we need to introduce a new block (in red in Figure 8), which is responsible of updating the feature database.

Feature database update

Ideally the feature database FP_{db} could be simply defined as the union of all feature sets FP_k associated to the previously aligned range images, each one brought in the absolute reference system through its own roto-translation matrix. However, such an approach would create plenty of redundancy in the database, since every corresponding feature (the ones on which we rely in order to assess the alignment between the views) would appear as replicas. A better approach would require us to realize an intersection

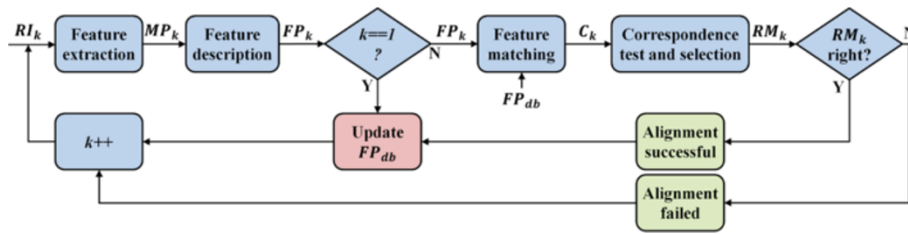


Figure 8 Block diagram representing the multi-view alignment system.

between the feature sets, in order to effectively curbing the database growth:

$$FP_{db} = \bigcap_{i=1}^{k-1} FP_i \quad (13)$$

To do so, for each feature set FP_k that needs to be inserted in the database FP_{db} we identify any corresponding appears in the same 3D position (that is, closer than $1/3$ of the feature signature radius). For each pair of concurrent features we select a single representative based on the similarity of the feature normal with respect to the acquisition direction. In fact, we assume that such features possess better signatures, since their neighborhood is usually less afflicted by occlusion issues. We also associate to each representative a *presence list*, that is a list of all the range images in which that feature appears: this list is exploited during the triplet verification phase (described in Section “Correspondence test and selection”) to determine on which range images the test has to be performed.

Global adjustment

After having successfully applied the described multi-view coarse registration technique to a given dataset, we have the possibility to complete the alignment process toward a high-precision object modeling with a fine alignment and/or global adjustment step. Considering the multiple view nature of our problem, a multi-view fine alignment technique should be used to prevent residual errors to propagate along the alignment path (e.g. closure problems for objects with cylindrical symmetry). Among several methods present in literature we used an approach we recently proposed [32] which is suitable to accurately align sets of high-resolution range images, also directly starting from the coarse aligned dataset. This is particularly suitable in cases a chained pairwise fine alignment (e.g. using ICP) is prevented by the absence of a chained path. Our approach is based on the ‘Optimization-on-a-Manifold’ framework proposed by Krishnan et al. [33], to which we contribute with both systemic and computational improvements. The original algorithm performs an error minimization over the manifold of rotations through

an iterative scheme based on Gauss–Newton optimization, provided that a set of exact correspondences is known beforehand. As a main contribution we relaxed this requirement, allowing to accept sets of inexact correspondences that are dynamically updated after each iteration. Other improvements were directed toward the reduction of the computational burden of the method while maintaining its robustness. The modifications we have introduced allow to significantly improve both the convergence rate and the accuracy of the original technique, while boosting its computational speed [32].

Acquired datasets

In order to compare each new building block of our pipeline in Section “Comparative tests”, as well as to evaluate both the pairwise and multi-view alignment performances in Section “Experimental results”, we consider a number of meaningful datasets that we acquired with a commercial high-resolution structured-light scanner (1280×1024 pixel CCD, i.e. potential 1.3M points/RI). The acquired datasets correspond to 14 scanned objects for a total of 300 range images, that is 286 RI pairs. They are shown in Figure 9, as they appear after successful alignment, with different colors associated to each range scan (naturally high color mixing due to scan inter-penetration is present). To make the pairwise and multi-view approaches fully comparable, among all the possible unchained acquisition paths a chained path has been determined and followed for each dataset. It is worth noting that for multi-view alignment a chained path can be considered as a regular element (with no special properties or distinction) of the set of unchained paths (according to the definition given in Section “Introduction”). In addition, we will also consider alternative unchained paths to further evaluate the multi-view pipeline. The considered objects are related to cultural heritage (statues and high-reliefs) and biomedical (orthodontic moulds) application fields, they have been chosen to represent well differentiated aspects and geometric properties, variegated feature dimensions, morphology and numerosity.

Objects size range from 50 up to 600 mm over their main dimension. Despite the acquisition paths aimed to minimize the number of scans while covering the entire

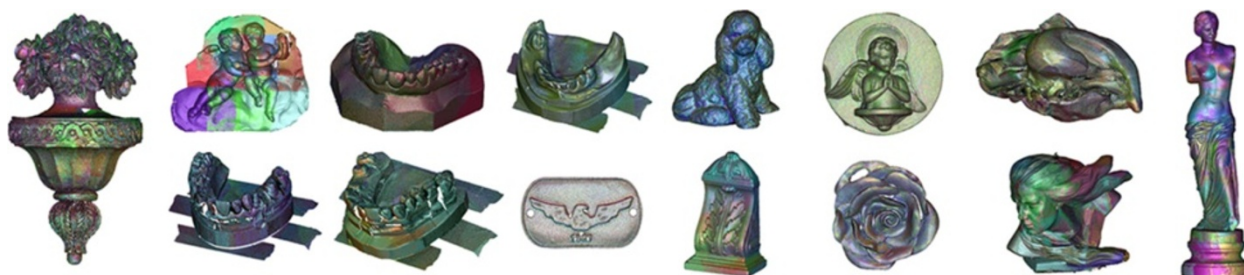


Figure 9 The test datasets: Decoration, Angels, Mould 1, Mould 2, Dog, Cherub, Dolphin, Mould 3, Mould 4, Platelet, Capital, Rose, Hurricane, Venus.

surface of interest, the amount of overlap between the images can vary a lot due to the implicit characteristics of the surface that is being acquired: if the surface is quite planar, as for the Angels dataset, the maximum overlap is limited to 30%, whereas for datasets particularly affected by occlusion phenomena, such as the Rose dataset, many images cover the same area from different viewpoint angles, thus increasing the region overlap up to 80%. However, in these cases, an higher overlap does not necessarily imply an easier alignment, since the views can be heavily affected by holes which may cause instability in the feature localization as well as a degraded reliability of the computed descriptors for the features close to the hole borders.

Comparative tests

The pipeline we proposed in Section “Pairwise alignment technique” may appear to be constituted by independent blocks, thus easily interchangeable with alternative ones. However, as anticipated in Section “Related work”, a certain degree of interdependency is present. In fact, our descriptor incorporates the saliency information, which is obtained during the multi-scale analysis performed during the extraction phase. A more subtle dependency regards the correspondence skim procedure: the number of correct correspondences that is obtained at the end of the matching block can be very limited, therefore the skim procedure has to be very reliable. This is the reason for which we came out with our own skim procedure rather than relying on heuristic approaches, which usually have more stringent assumptions with regard to the number of inliers within the correspondence set. These considerations are to be taken into account when trying to compare the performance of each pipeline block: substituting a given technique for another one, extrapolated from a different framework, may result in performance degradation. With these considerations in mind, we now propose a number of comparisons of our pipeline stages with respect to other works presented in the literature. We tried to identify solutions that may individually fit into

our pipeline in order to reduce performance degradation effects due to technique unsuitability.

In Section “Feature extractor and descriptor comparison” we evaluate both the feature extractor (proposed in Section “Feature extraction”) and descriptor (introduced in Section “Feature description”) with respect to similar approaches on a number of different scenarios we devised, while in Section “Correspondence test technique comparison” we evaluate our correspondence skim procedure (described in Section “Correspondence test and selection”) with respect to other solutions by comparing both precision and computational efficiency. All the comparisons presented in this section have been performed on the same platform described in Section “Pairwise alignment results”.

Feature extractor and descriptor comparison

In order to assess the performance of the extraction and description blocks of our pipeline, we devised a number of test scenarios. Hereafter we state what is the objective we pursue for each block; then we describe the test scenarios.

In order to compare our feature extraction technique with respect to the others, we aim to assess the capability to detect salient points in the same position, regardless to the possible variations that the surface may undergo due to various effects. We therefore devised three different scenarios in which we assess the feature repeatability with respect to: (1) noise added to the surface; (2) holes carved in the data; (3) acquisition viewpoint variation.

For the feature description stage, our objective is to assess how well our descriptor compares with others. To do so, given two aligned range data we extract features from both of them, match them, and count the ones that are correct (i.e. the ones for which the Euclidean distance is below a given threshold). Out of all possible correspondences, we retain the ones that present the best score values in the set C_{ab} , which size Q is set to 150, and count the correct correspondences that survived the selection step. Intuitively, the better the descriptor is, the greater its correspondence survival rate will be. Similarly to the previous assessment, we plan the same scenarios in which we

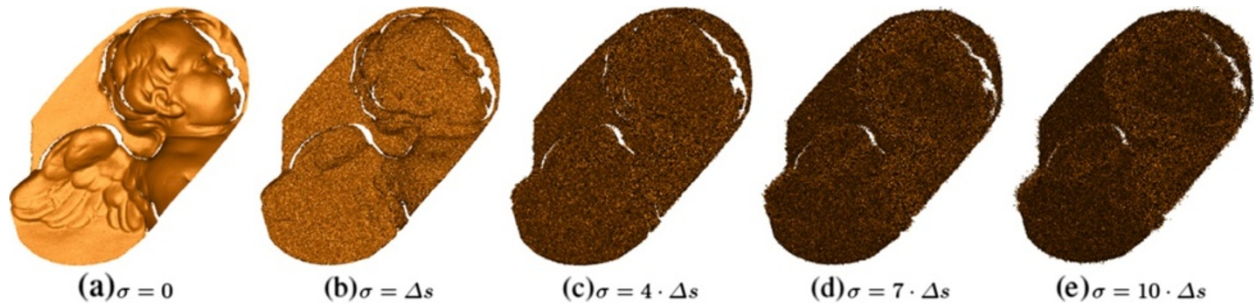


Figure 10 Angels particular with added noise at different σ .

measure the correspondence survival rate. Following, we describe in detail each test scenario.

Noise scenario: for a given range image RI we generate a number of replicas RI_{noise}^k , with $k \in [0, 10]$, where each replica is smeared with Gaussian noise with zero mean and standard deviation $\sigma = k \cdot \Delta s$, where we set Δs equal to the average grid spacing. In order to collect enough information to infer meaningful statistics, for this scenario six different datasets were used, for a total of 105 test images. In Figure 10 an example of noisy data is shown.

Hole scenario: for a given range image RI we generate a number of replicas RI_{holes}^k , with $k \in [1, 4]$. For each replica we add holes which size varies randomly following a Gaussian distribution with mean $\mu = (10 \cdot k \cdot \Delta s) - 5$ and standard deviation $\sigma = (5 \cdot k \cdot \Delta s) + 10$, where Δs is again the average grid spacing. Similarly to the previous scenario, the same datasets were used during the comparison. Figure 11 reports an example of carved data.

Viewpoint variation scenario: in this particular setup, we acquired a planar high-relief object surface (the represented in the Angels dataset) from a number of different viewpoints. We acquired 33 range images varying the acquisition angle with respect to the plane normal within a range of $[-45^\circ, 45^\circ]$ at step angle of almost 3° . The image acquired at 0° is then tested with all the other scans separately and, for each of these image pairs, useful test statistics are gathered, according to the kind of

comparison that needs to be performed (see below). Due to occlusions and limited acquisition volume, the overlapping area between the two scans will vary. To provide a fair comparison, we estimate the overlapping area of each scan pair and apply proper normalization factors. This scenario is of particular interest for our application, since relative viewpoint variations are very likely to occur during object acquisition. In Figure 12 we display the surface, acquired from different viewpoints, used as a test for this scenario.

Feature extractors comparison

In this section, we compare the feature extraction approach described in Section “Feature extraction” with respect to Lee et al. [20], here referred as *Lea*, Castellani et al. [22], *Cea*, and Bonarrigo et al. [6], *Bea*. It is worth to remember that the proposed approach is a variation of *Cea*, and that both *Lea* and *Cea* were originally conceived to work with mesh datasets, therefore they have been adapted here to work with range images.

In the noise scenario, we assess the feature repeatability with respect to synthetic noise added to the data. To do so, for each pair of (prealigned) images $(RI, RI_{\text{noise}}^k)$ features are extracted and their position is matched in order to determine the number of repeating features. Since on data RI_n^0 no noise is added, the number of repeating features rF_{noise}^0 will coincide with the number of extracted features. Therefore, for $k \neq 0$, rF_{noise}^k will always result

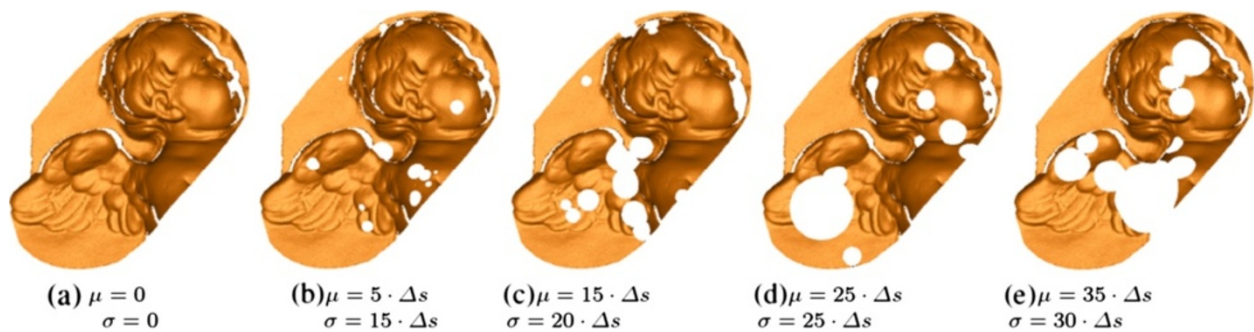


Figure 11 Angels particular with carved holes at different μ and σ .



Figure 12 The planar surface taken from different acquisition angles θ .

less than rF_{noise}^0 . In Figure 13 we represent the feature repeatability for increasing noise levels as the percentage $100 \cdot rF_{\text{noise}}^k / rF_{\text{noise}}^0$.

The graph shows that the proposed approach performs best, while *Lea* approach fares slightly worse. The technique we proposed in [6] is around 10% worse, while *Cea* shows a performance decrease of 20%. The main critical aspect here resides in the fact that noise corrupts the reliability of the surface normal estimation. This is detrimental in *Cea* where original normals are used to improve DoG saliency computation. This problem is mitigated in *Bea* and in the proposed modified *Cea* approaches, where we recompute the normal field on the Gaussian filtered surfaces. The original *Lea* approach does not employ normals, therefore it is more robust to noise effects. Interestingly, even using (smoothed) normal information, the proposed method is more robust than *Lea* to noise.

In the hole scenario, we assess the feature repeatability with respect to holes carved in the data. To perform such evaluation we first check the number of features extracted from RI_{holes}^k , which corresponds to the number of repeating features $rF_{\text{holes}}^{k,k}$ that one would obtain by matching RI_{holes}^k against itself. Then the (prealigned) image pair $(RI, RI_{\text{holes}}^k)$ is matched and the number of repeating features $rF_{\text{holes}}^{0,k}$ is counted. In Figure 14 we represent the feature repeatability for different hole levels as the percentage $100 \cdot rF_{\text{holes}}^{0,k} / rF_{\text{holes}}^{k,k}$.

Again, the graph shows that the proposed approach performs best. *Bea* approach performs 4% worse, while *Lea* and *Cea* perform around 12% worse. Good performance of the *Bea* solution are related to its inherent robustness with respect to deformations of the DoG maps induced by unbalanced point density on range images [6] due to the abnormal quantity of borders and holes present.

For the viewpoint scenario, we report the feature repeatability rate with respect to variations of viewpoint angle. For a given pair of prealigned scans $(RI_{\text{vpoint}}^0, RI_{\text{vpoint}}^\theta)$, we determine the number of repeating features $rF_{\text{vpoint}}^{0,\theta}$ and estimate the overlap ratio between the views $oR^{0,\theta}$. As already stated in Section “Feature extractor and descriptor comparison”, since the overlapping area between the two scans is likely to vary, we apply a normalization factor to the feature repeatability rate, which is calculated as follows: $100 \cdot rF_{\text{vpoint}}^{0,\theta} / (rF_{\text{vpoint}}^{0,0} \cdot oR^{0,\theta})$. In Figure 15 the repeatability rate is displayed with respect to viewpoint angle θ .

Once again, the proposed approach performs best, while *Bea* approach obtains slightly inferior performance. In the angle interval $[-30^\circ, 30^\circ]$ *Lea* approach performs 30% worse, while *Cea* falls around 50% under the proposed one. For angle values greater than $\pm 30^\circ$, the performance gap decreases, reaching a repeatability rate of 25% for the proposed approach as well as *Bea*, and virtually no repeating features for *Lea* and *Cea*.

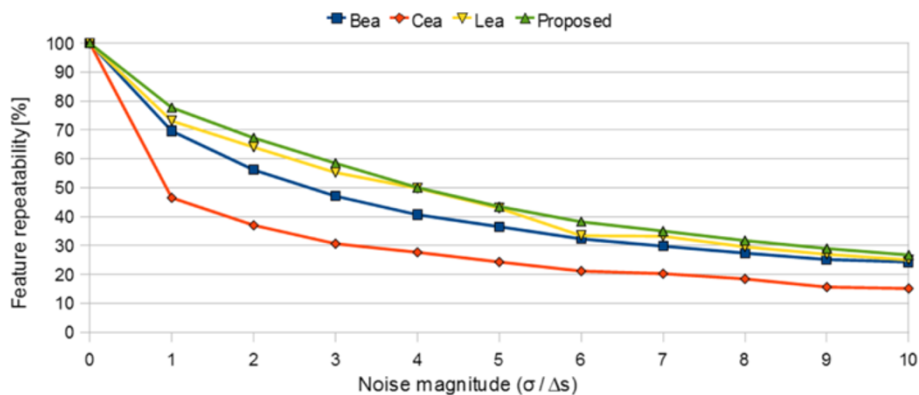


Figure 13 Feature repeatability rate with respect to noise.

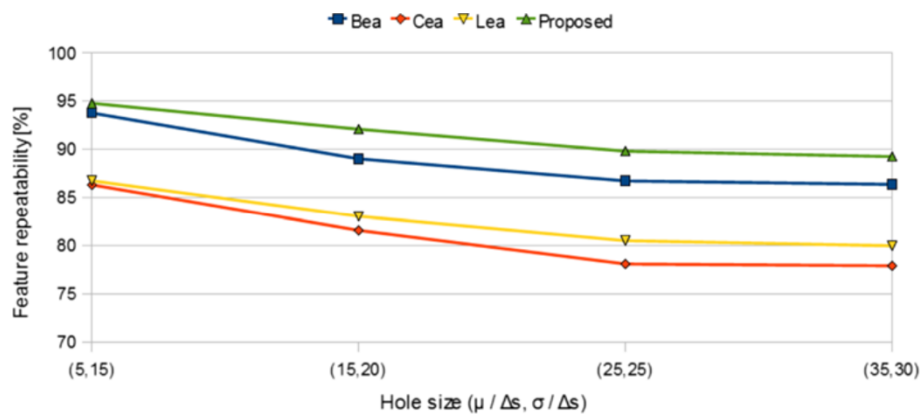


Figure 14 Feature repeatability rate with respect to holes.

In conclusion, our tests have demonstrated that the proposed extractor outperforms the other techniques with respect to noise, holes and variation of the acquisition viewpoint. The approach we first introduced in [6] turned out to perform slightly worse than the proposed one for holes and viewpoint variation, and to be vulnerable with respect to added noise. A few considerations need to be done with regard to the performances obtained by *Lea* [20] and *Cea* [22] approaches. These approaches were both originally introduced for meshes, which implicitly smooth the data with respect to range images and point clouds. Moreover, depending on the meshing algorithm employed, small holes may be closed automatically, or no holes may be present at all (for implicit surface reconstruction approaches, such as [34]). The low performances obtained for the noise and hole scenarios are therefore understandable, nevertheless their modest results for the viewpoint variation scenario remain a critical issue for our particular application field.

Feature descriptor comparison

In the present section we assess the correspondence survival rate of our feature descriptor (introduced in Section “Feature description”) with respect to the popular Spin Images approach, introduced by Johnson [30]. Since we apply Spin Images for partial view range data matching, we have to face the problem described in [35] (for the case of model recognition in cluttered scenes) about the determination of a good tradeoff between distinctiveness and robustness. The former is favored by global or wide field Spin Images, while robustness to view changes (in this case due to the nature of the data and of the addressed problem) can be improved by adopting localized (short range) descriptors. Therefore we first addressed this tradeoff by finding the optimal Spin Image window size for our data: we experimentally found that a window size of 15 and a bin size equal to the average point spacing gave the best alignment results on all the test datasets. Being calculated in a multi-scale framework,

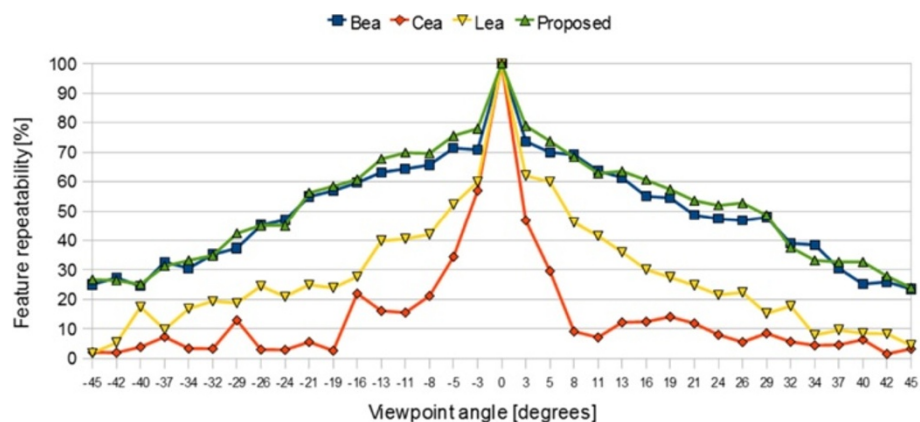


Figure 15 Feature repeatability rate with respect to viewpoint variation.

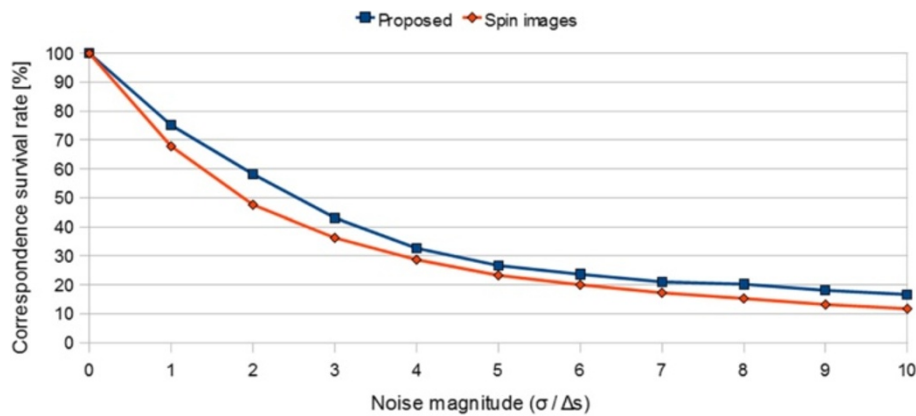


Figure 16 Correspondence survival rate with respect to noise.

the actual average extent of the Spin Images window with respect to the original range images corresponds to $r \cdot p \cdot 15$ points, depending on the scale parameter r and the preemptive subsample parameter p . Moreover, since we are interested in efficient feature-based reconstruction, similarly to what we do with our descriptor we only compute Spin Images on MP obtained from the extraction phase.

In the noise scenario we assess the correspondence survival rate with respect to synthetic noise added to the data. To do so, for each pair of (prealigned) images (RI, RI_{noise}^k) all exact correspondences are counted (we can do this because the image pair is prealigned), and their number recorded as eC_{noise}^k . Then, similarly to what we would do if the images were unaligned, the entire correspondence set is ranked and skimmed so that only the best 150 correspondences are retained, and the survived exact correspondences are counted and registered as sC_{noise}^k . In Figure 16 we represent the correspondence survival rate as the percentage $100 \cdot sC_{noise}^k / eC_{noise}^k$.

The graph shows that the two descriptors behave similarly with respect to noise added to the data, with our approach performing a bit better, especially for lower noise values.

In the hole scenario, we assess the correspondence survival rate with respect to holes carved in the data. To perform such evaluation, we first check the number of survived correspondences $sC_{holes}^{k,k}$ between RI_{holes}^k and itself. Then the (prealigned) image pair (RI, RI_{holes}^k) is matched and the number of survived correspondences $sC_{holes}^{k,k}$ is evaluated. In Figure 17 we represent the correspondence survival rate for different hole levels as the percentage $100 \cdot sC_{holes}^{0,k} / sC_{holes}^{k,k}$.

In the graph we can see how the Spin Images performance decreases faster than the proposed descriptor for bigger holes. Considering the Spin Image descriptor itself this is not surprising, since if a part of the data is missing, this will have a (proportionally) greater effect on the Spin Image histogram rather than on our grid descriptor, where only a subset of the sectors will be affected.

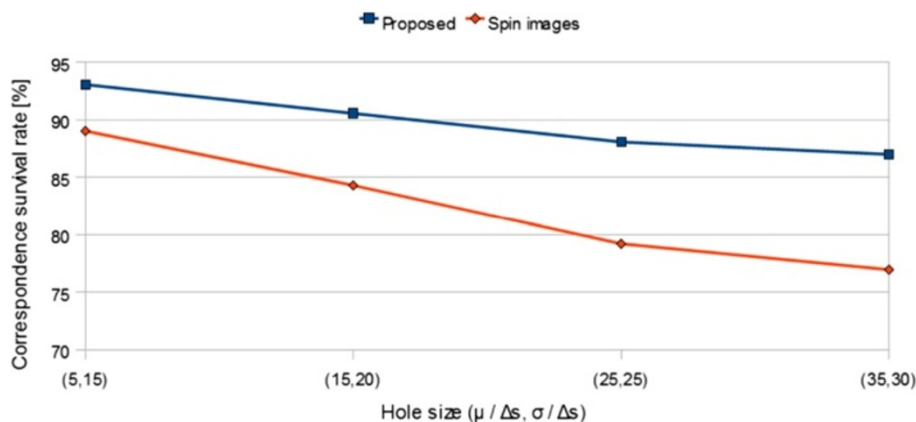


Figure 17 Correspondence survival rate with respect to holes.

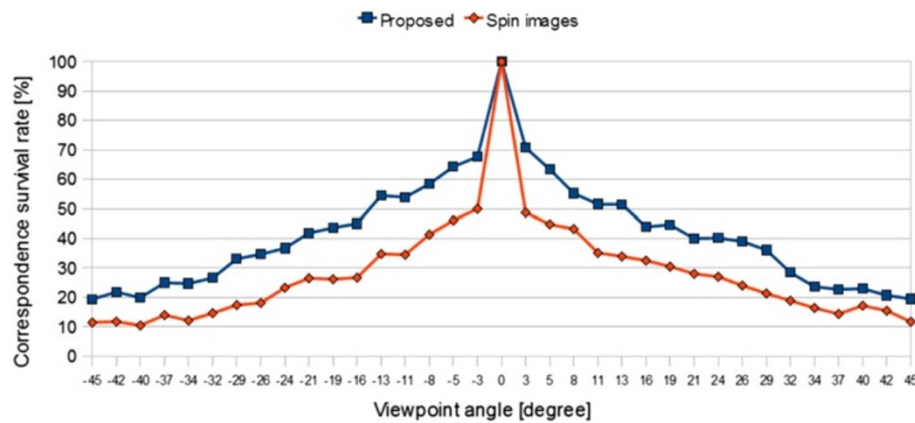


Figure 18 Correspondence survival rate with respect to viewpoint variation.

For the viewpoint scenario, we determine the correspondence survival rate with respect to variations of viewpoint angle. For a given pair of prealigned scans $(RI_{\text{vpoint}}^0, RI_{\text{vpoint}}^\theta)$, we determine both the number of survived exact correspondences $sC_{\text{vpoint}}^{0,\theta}$ and the overlapping ratio between the views $oR^{0,\theta}$. We then apply a normalization factor to the correspondence survival rate, which is calculated as follows: $100 \cdot sC_{\text{vpoint}}^{0,\theta} / (sC_{\text{vpoint}}^{0,0} \cdot oR^{0,\theta})$. In Figure 18 the survival rate is displayed with respect to viewpoint angle θ .

In the graph we can see that, within an angle interval of $[-30^\circ, 30^\circ]$, the proposed descriptor performs 20% better than Spin Images, while for greater angle values the performance gap decreases up to 10%. This outcome could have been foresaw: viewpoint variations are likely to introduce holes on the surface due to occlusion. As we demonstrated for the previous scenario, Spin Images descriptors appear to be more sensitive to holes than our own, and these results confirm it.

We can conclude that both the descriptors perform similarly with respect to the noise scenario, while the proposed descriptor outperforms the Spin Images for the holes as well as the viewpoint variation scenarios. With regard to the time required for generating the descriptors, we estimated that describing a single Spin Image (with a window size of 15) costs us 0.61 ms, while our descriptor only takes 0.34 ms to compute, thus demonstrating the fact that our descriptor is extremely fast to compute. Although such difference is small in absolute value, it becomes significant when considering that the set of MP is usually constituted of hundreds of elements.

Correspondence test technique comparison

For the correspondence skim procedure, our aim is to assess the capability of different techniques to detect a correct triplet in the correspondence set C_{ab} , as well as

the time required to do that. We compare the approach we introduced in Section “Correspondence test and selection” with respect to the RANSAC approach proposed by Fischler et al. [12] as well as its descendant PROSAC introduced by Chum et al. [31]. PROSAC should perform better when, as in our case, a correspondence ranking criterion is available. In fact, while RANSAC treats all correspondences equally (by choosing random sets), PROSAC works its way progressively down the top-ranked ones. Since the ranking we obtain through the feature matching procedure is fairly good (for example, in the Angels dataset the inlier ratio is on average around 70% for the first 25 top positions), in our case the PROSAC requisite should be satisfied. In order to boost the computational speed for these two techniques, which is heavily dependent on the inlier rate of the correspondence set, a dynamic estimation of such rate is performed for each set while running the techniques. This allows the algorithms to terminate faster in case the inlier rate is superior than in the worst case [36].

In order to perform the comparison, we apply the pairwise alignment pipeline over six different datasets, for a total of 99 RI pairs. During the process we keep track

Table 1 Alignment errors for each skim technique

Dataset	Number of RI pairs	Number of errors for proposed	Number of errors for RANSAC	Number of errors for PROSAC
Angels	7	0	0	0
Teeth	7	0	0	0
Dog	13	0	2	0
Dolphin	19	0	1	0
Capital	22	0	0	0
Hurricane	31	1	2	1
Total	99	1	5	1

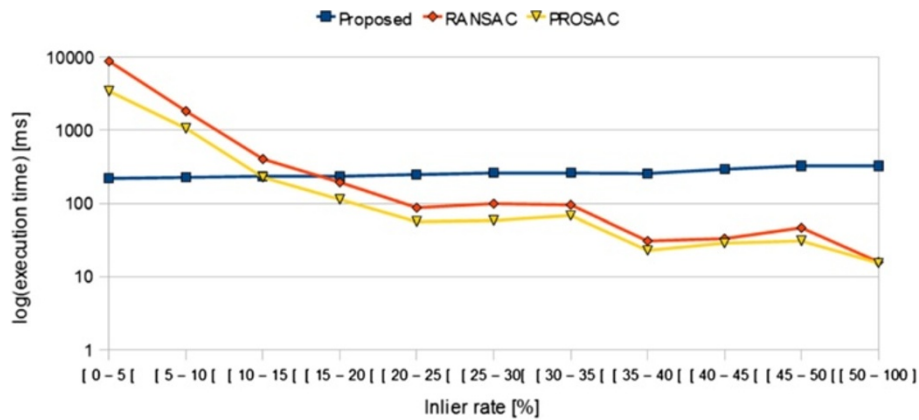


Figure 19 Execution times for each skim technique.

of the inliers rate within the set C_{ab} , as well as the time required by each technique to determine their best correspondence triplet within C_{ab} . Once the alignments are terminated, we visually inspect the alignment outcome and count the number of errors for each technique, reported in Table 1. Given the same correspondence sets C_{ab} , both the proposed approach as well as the PROSAC performed best, with a single error. On the contrary, the RANSAC method introduced a considerable amount of errors. In Figure 19 we also present in logarithmic scale the execution times required by each technique with regard to the inliers rate. As expected, the time required by RANSAC as well as PROSAC grows exponentially with the number of outliers in the correspondence set C_{ab} , ranging from 15 up to 8800 milliseconds for RANSAC, or 3400 milliseconds for PROSAC. On the contrary, the proposed skim procedure has a balanced dynamic, ranging from a minimum of 220 up to 330 milliseconds. It turns out that our approach is up to 40 times faster than RANSAC and 15 times faster than PROSAC when the inlier rate is low, whereas it gets down to 20 times slower for higher inlier

rates, with curve crossing rate of 16% for RANSAC and 12,5% for PROSAC. Such analysis may seem to favour the PROSAC technique, however before drawing any conclusion we also need to take into account the inlier rates distribution, which we estimated from the six test datasets (99 RI pairs). As we can see in Figure 20, the inlier rate is usually very low: for more than half cases, its value is below 15%, due to the nature of the partial view alignment problem we face. In fact, any variation of the relative position between the scanner and the object is likely to cause a variation in the overlapping area between successive scans, with the possible creation of holes due to occlusions, as well as variations on the distribution of 3D points on the surface. All these factors concur in lowering the number, as well as the similarity of repeating features, thus causing an overall reduction of the inlier rate. Taking into account the inlier rate distribution we estimated the weighted average execution time for the three skim techniques. It turns out that our approach is the fastest, since it requires 240 milliseconds to compute its best triplet. RANSAC needs 1460 milliseconds to obtain its guess,

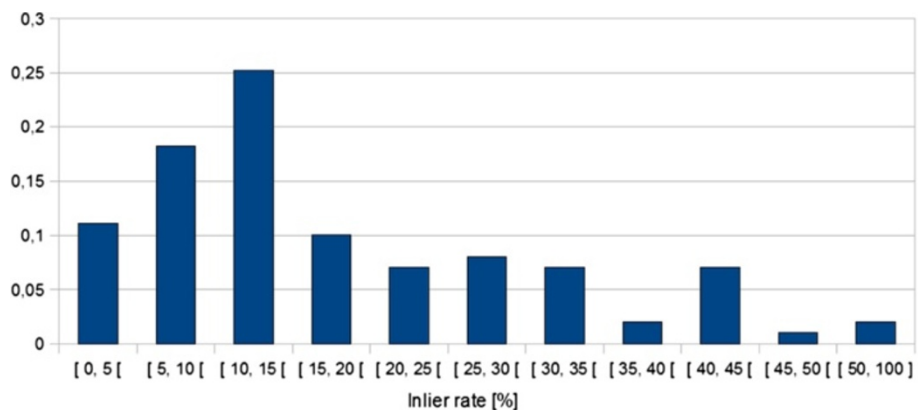


Figure 20 Distribution of inlier rates for the test datasets.

while PROSAC requires 660 milliseconds. In conclusion, considering the results of Table 1 as well as the average execution times, we can conclude that both our technique as well as PROSAC obtained the best alignment performances, while our approach is almost 3 times faster than PROSAC, given the inlier distribution rates of the test datasets.

Experimental results

We evaluated both the pairwise registration and the complete multi-view alignment pipeline presented in Section “Pairwise alignment technique ” and Section Multi-view alignment pipeline, respectively through a series of tests performed on the datasets presented in Section “Acquired datasets”. Based on the feature size of each object, we found suitable to define two different parameter configurations, which we address as the standard feature (*std*) and the small feature (*sml*) configurations. The *std* one is configured as follows: a preemptive factor 2 subsampling of the range image, feature extraction on 3 octaves, one saliency map for each octave and a Gaussian kernel size set to 4, feature signature with a number of angular sectors equal to 36; whereas the *sml* configuration consists in: no preemptive subsample, 3

octaves, one saliency map for each octave, a Gaussian kernel size set to 3 and a number of angular sectors equal to 18.

Dataset characteristics are presented in the first 4 columns of Table 2, where they are listed in ascending order in terms of number of RI pairs. Average number of valid points per RI and configuration parameters are also provided.

Pairwise alignment results

In order to evaluate the pairwise approach, we executed the feature-based pairwise alignment procedure of Section “Pairwise alignment technique ” on each RI pair of each dataset and counted the number of successful alignments. At first, we visually check the resulting alignment, and count the number of (manifestly) correct and wrong occurrences. For dubious cases we run an ICP to assert whether the coarse alignment is sufficient or not for the ICP to converge. Quantitative results are presented in columns 5 and 6 of Table 2. Considering the first 14 datasets, the technique demonstrated to be very robust in that it correctly aligned 96.5% of the RI pairs. Further analysis performed over the few unaligned pairs concluded

Table 2 Experimental results summary

Dataset	Number of RI pairs	Avg. Number of points/RI	Param. config.	Pairwise alignment		Multi-view alignment			
				Number of RI pairs aligned	Avg. exec. time/RI [s]	Number of clusters created	Avg. exec. time/RI [s]	Initial align. error [μ m]	Final align. error [μ m]
Angels	7	1M	std	7	3.6	1	4.9	62	27
Mould 1	7	410k	std	7	2.0	1	2.6	39	14
Mould 2	9	400k	std	9	1.7	1	2.2	66	11
Mould 3	9	370k	std	9	1.5	1	1.7	103	11
Mould 4	9	360k	std	9	1.6	1	2.0	165	12
Platelet	11	80k	sml	11	1.5	1	1.8	28	9
Dog	13	130k	sml	13	2.5	1	5.0	32	16
Cherub	19	650k	sml	18	1.3	1	1.6	149	20
Dolphin	19	410k	std	19	2.0	1	4.1	136	33
Capital	22	760k	std	22	3.1	1	6.6	127	23
Rose	23	170k	sml	23	2.8	1	6.1	67	13
Hurricane	31	690k	std	30	2.9	1	9.0	188	35
Decoration	47	510k	std	39	2.7	2	10.3	–	–
Venus	60	835k	std	60	3.1	1	12.1	152	17
Dog ^a	13	130k	sml	7	2.4	1	5.1	41	16
Capital ^a	22	760k	std	11	2.9	1	7.5	176	23
Hurricane ^a	31	690k	std	17	2.9	1	9.6	255	35
Carter ^b	92	550k	std	64	2.5	18	16.8	–	–

^aAcquisition path for these datasets does not abide by the overlap constraint;

^bParticularly featureless dataset.

that main causes for failure were due to either an insufficient overlapping area (that is, close to the lower bound of 20%), or particularly featureless areas. A visual example of the output produced by our pairwise alignment technique is shown in Figure 21.

Computational performance (column 6) are related to a C++ implementation and run on a PC equipped with an Intel 2.4 GHz dual-core processor and 4 GB of RAM; time required for disk loading/saving of range data is excluded. It is important to note that the code has not been fully optimized for parallel execution yet, hence there is room in this sense for further improvements of time performance. Computational performance shows an average alignment time of 2.5 s. As an example, time breakdown for the Hurricane dataset is distributed as follows: 57% for feature extraction, 1% for feature description, 32% for

feature matching, 8% for correspondence skim and roto-translation estimation. Lightness of our feature signature is testified by the fact that the feature description step only takes 1% of the total time. The two main factors that influence computation times are the number of points per range image to be processed, and the number of features detected over each image. In the “worst case” (that means, images close to 1 million of points and many features detected at all scales), alignment time reached a maximum of about 4 seconds. Comparisons of computational speed with respect to the literature is somehow difficult to infer since (1) not every work declare computational speed, (2) only isolated blocks of the alignment chain are usually considered (e.g. feature extraction) instead of the entire pipeline, (3) hardware obsolescence. Nevertheless, our computational time is at least one order of magnitude

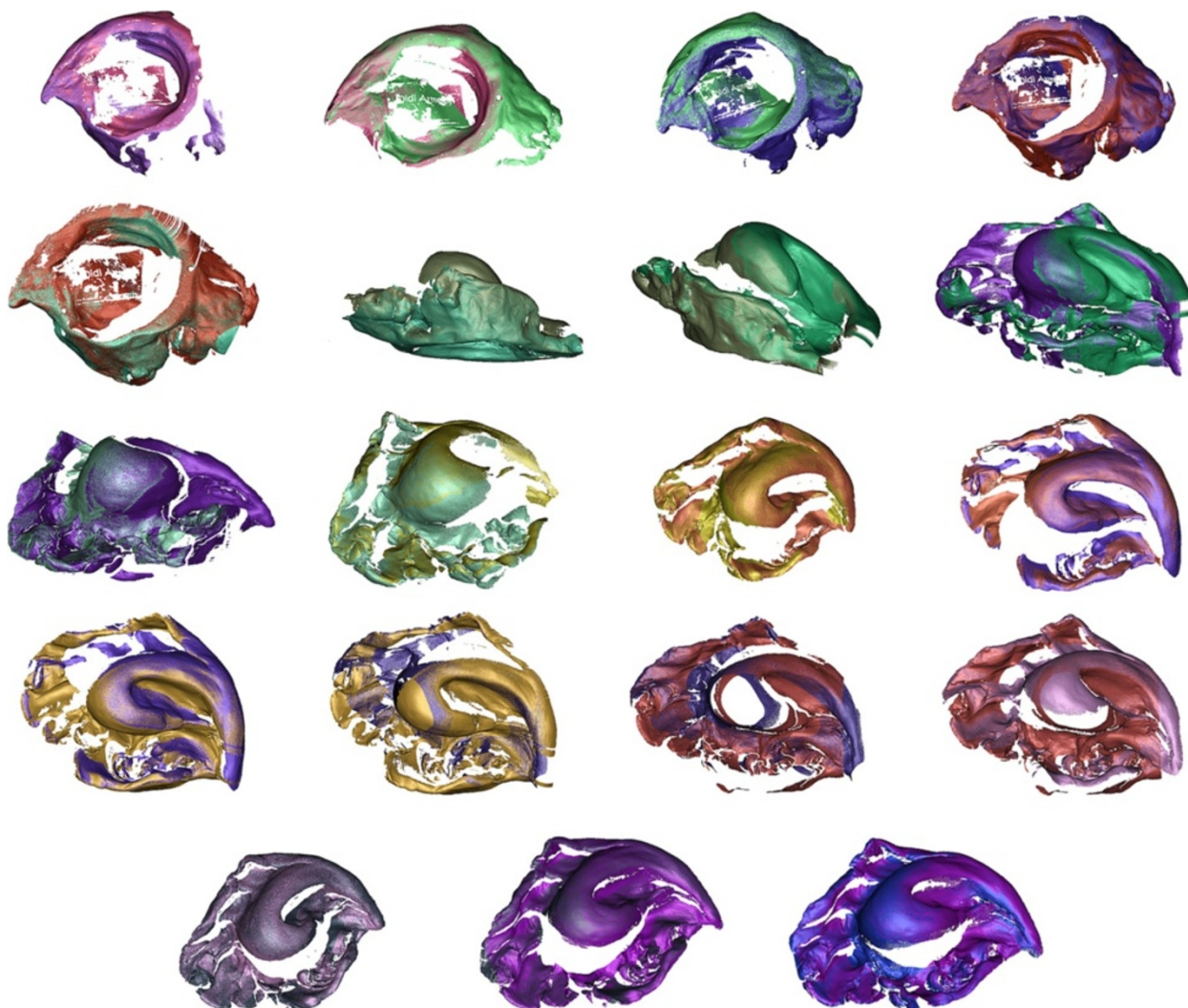


Figure 21 Pairwise alignment results for the Dolphin dataset.

(comprising hardware obsolescence compensation) under the times declared in the related works [2,19,20,22].

Multi-view alignment pipeline results

For the multi-view alignment pipeline, our aim is to progressively complete the object coverage with an unconstrained acquisition procedure. In order to do so, we match the features extracted from the current image with respect to the entire feature database. Therefore, whenever a single alignment error occurs, there is a chance that erroneously aligned views may attract more range data, creating clusters of correctly aligned views. This is the reason for which, in order to assess the multi-view alignment performance, in Table 2 col.7, we state the number of different clusters created, rather than counting the correct image pairs. If the multi-view alignment succeeds in its task, a single cluster will result, otherwise more clusters will be created.

In this case we consider as correct an alignment where the current range image is attached correctly to an existing cluster (that is, we consider as erroneous only the alignment for which a new cluster is created, rather than all the other images which may attach to it correctly). To verify correct alignments we use the same criteria stated in Section “Pairwise alignment results”. Quantitative results are presented on the right side of Table 2. Again, the technique demonstrated its robustness and successful object reconstruction is reached in almost all cases. With respect to the pairwise approach, the multi-view approach solved the alignment failures of the Cherub and the Hurricane datasets by exploiting the cumulative information available. A single error occurred on the Decoration dataset, generating two distinct image clusters, as shown in Figure 22. In terms of correctly aligned RI pairs, our multi-view alignment reaches a 99.7% performance.

As already stated, a major limitation for the chained pairwise alignment is that it can succeed in obtaining a correct reconstruction only if the range data sequence follows an adequate acquisition path. To exemplify this, and to assess the path-independence of the multi-view approach, we performed a test on alternative acquisition paths defined on the Dog, Capital and Hurricane datasets (see the rows of Table 2 marked by a single asterisk). For these datasets the acquisition path has been rearranged (respecting the assumption of partial overlap with at least one of the previous range images in the path). In such unconstrained (unchained) condition the pairwise alignment cannot perform correctly (in fact only about half of the total RI pairs were correctly aligned along the alternative paths) while the multi-view alignment still performs correctly.

The last two columns of Table 2 present an estimation of the root mean square error among the aligned



Figure 22 The Decoration clusters: a single error caused the creation of two distinct clusters of correctly aligned views.

views, prior and after performing the global optimization-on-a-manifold adjustment introduced in Section “Global adjustment”. The alignment was performed only on the datasets which were correctly reconstructed (all except the Decoration and the Carter). As can be seen, all the (coarsely aligned) datasets have been aligned finely, and the residual error (generated by scanner miscalibration and noise associated to the acquisition process) is never greater than $35\mu\text{m}$ for an object which main diagonal is 40 cm long, thus a very good alignment. This demonstrates that the coarse alignment obtained with our approach is good enough to bring the multi-view fine alignment stage to converge to the correct solution. Interestingly, the three datasets that present an alternative acquisition path result in the same residual error after the global adjustment phase, which is another confirmation of

the fact that the obtained coarse alignment is good enough for the successive alignment stages to converge toward the optimal solution. In Figure 23 the Venus dataset is shown for three alignment stages: (a) is the initial image condition for the dataset; (b) shows the alignment obtained after executing our multi-view pipeline; (c) represents the dataset after executing our global optimization method [32].

As for limitations, our method presents essentially two: a) performance degradation for featureless objects, and b) complexity increase (although less than linear) with the number of views. For point (a), the last row of Table 2 (the one with double asterisk) reports the alignment performance for the additional dataset Carter. This is a mechanical piece, with smooth planes and holes carved in, thus a particularly featureless dataset. For this dataset the alignment performance dropped considerably both for the pairwise and the multi-view reconstruction. In Figure 24 the dataset is shown, as well as two examples for successful alignments and one failure. Regarding point (b), the benefits of multi-view alignment come at a cost in terms of computational performance which can be easily realized by evaluating how the time breakdown varies (again, estimated for the Hurricane dataset): 19% for feature extraction, negligible feature description time, 78% for

feature matching, 3% for correspondence skim and roto-translation estimation. As the feature database grows, the time required for matching correspondences between the current dataset FP_k and the database database FP_{db} grows linearly (actually less than linearly, due to the removal from the database of duplicated features) with the number of views. This cannot be easily inferred Table 2, since each dataset has a different amount of features (for example, the Cherub dataset is processed in less than one third of the time required for the Dog, which is smaller in RI size but richer in features). Feature organization solutions, such as clustering or bags of words approaches [18], are possible and should be studied to cope with this limitation for bigger datasets. Possible solutions to this problem could also stem from the application domain, for example: (1) try a pairwise alignment before executing the alignment on the entire database (in fact, in many practical cases unchained paths are made of chained fragments, this could therefore save a lot of computation time); (2) distribute the feature database in a feature space, so that the matching for each new feature could be performed on the subset of features which are closer to it in the feature space (3) try to clear out parts of the database in order to lighten the matching by discarding those features that belong to already well covered zones. Alternative

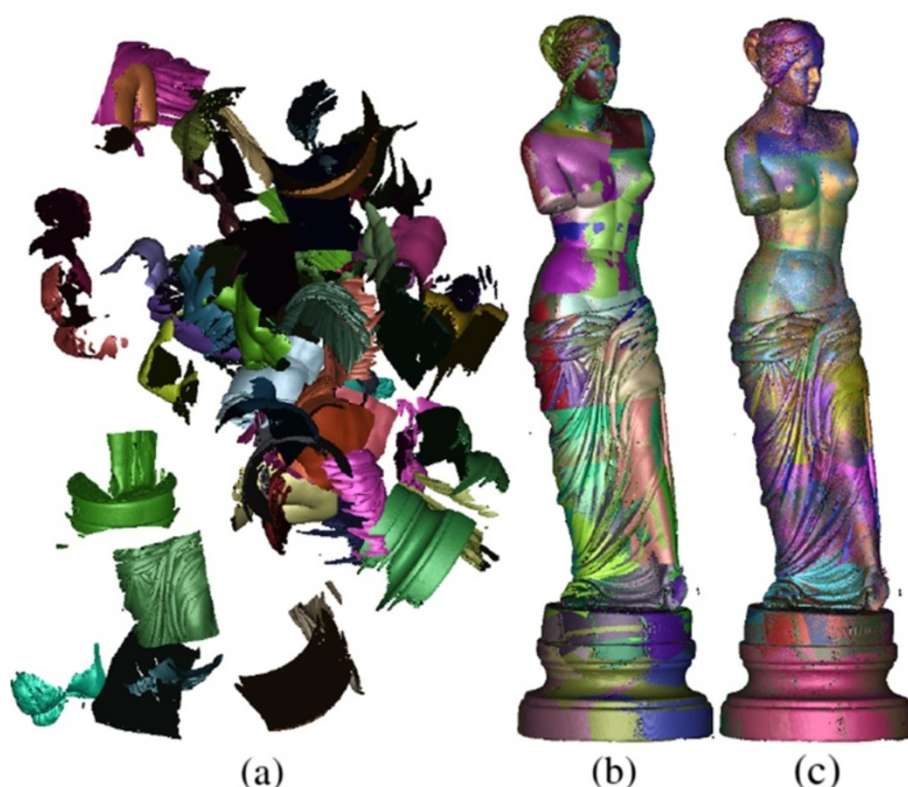


Figure 23 The Venus alignment phases: (a) initial condition, (b) after coarse multi-view alignment, (c) after global adjustment.

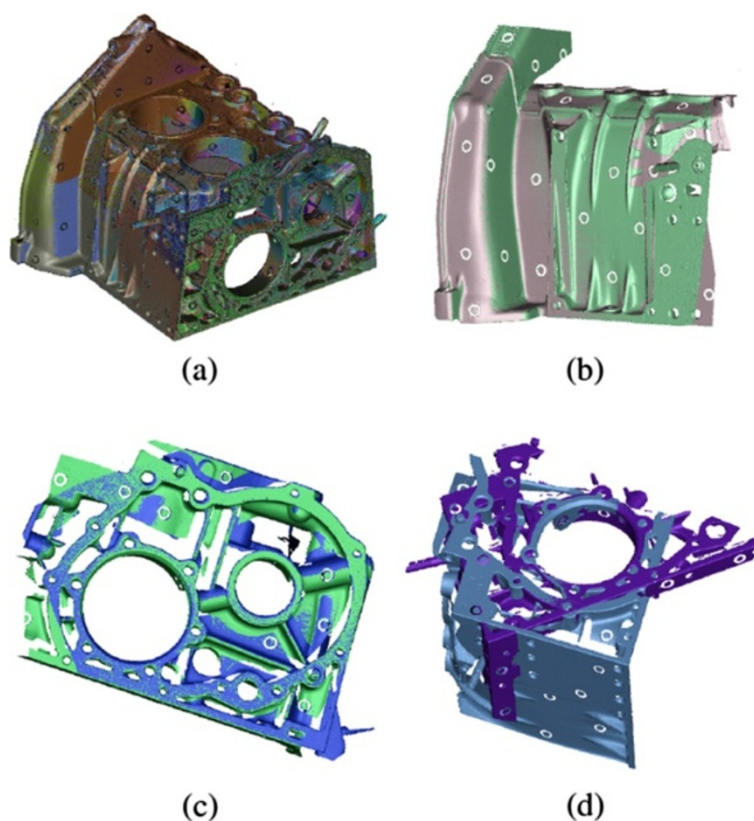


Figure 24 (a) The Carter dataset: (a) global picture, (b) and (c) two successfully aligned views, (d) an alignment failure example.

solutions can be devised according to application-driven requirements.

Conclusions

In this work we presented a fully automatic, fast and feature-oriented 3D alignment pipeline for high-quality object modeling from an unchained acquisition of dense range scans. The proposed approach takes as input a set of range images ordered according to a suitable user-defined acquisition path, then (1) extracts a set of feature points; (2) describes them through information extrapolated from the local surface; (3) matches the set with respect to the database of feature points associated to any previously aligned range image; (4) identifies the reliable matches and (5) tries to compute a roto-translation matrix out of such correspondences. The multi-view framework allows object alignment under the most reasonable assumption that each new range image must possess an overlap with respect to any other previously aligned range data, thus weakening the requirement of having an overlap between each sequential pair of range images. Several comparisons have been performed to evaluate the performances of each block of the pipeline we introduced with respect to competitor solutions, for a variety of different use scenarios.

The pipeline performance has been evaluated on a group of 14 datasets, for a total of 300 highly resolved range scans. Obtained results show a high degree of robustness and reliability of the technique, and are relevant in terms of improving the usability and the handiness of modern 3D scanners by allowing interactive (on-the-fly) object alignment and therefore fast modeling. In few cases clusters of correctly aligned views appeared, mostly due to possible inter-object feature similarities. However, the detection of this kind of problem can be rendered automatic or semi-automatic and deferred at an implementation level according to specific application requirements. A major issue that we observed is the increased amount of computational time due to the matching with a database of features. Solutions can be devised by using feature space organization approaches. However, this usually does not impair an interactive usage of the scanner, except for very big datasets. In conclusion, for the greatest part of the considered real life datasets, the proposed method guarantees competitive performance for a robust, fast, accurate and automated object alignment, consequently boosting the usability of high-end 3D scanners.

Competing interests

The authors declare that they have no competing interests.

Received: 1 June 2011 Accepted: 1 June 2012
Published: 18 July 2012

References

1. D Huber, M Hebert, Fully automatic registration of multiple 3D data sets. *Image Vis. Comput.* **21**(1), 637–650 (2003)
2. Q-X Huang, H Pottmann, *Automatic and robust multi-view registration, Geometry preprint 152*. (Vienna University of Technology, Dec 2005)
3. N Gelfand, NJ Mitra, L Guibas, H Pottmann, in *Symp. on Geom. Processing*. Robust global registration, (2005)
4. F Bernardini, H Rushmeier, The 3D model acquisition pipeline. *Comput. Graph. Forum.* **21**(2), 149–172 (2002)
5. Kaick van O, H Zhang, G Hamarneh, D Cohen-Or, A survey on shape correspondence. *Comput. Graph. Forum.* **30**(6), 1681–1707 (2011)
6. F Bonarrigo, A Signoroni, R Leonardi, in *ICCV*. A robust pipeline for rapid feature-based pre-alignment of dense range scans, (2011)
7. J Salvi, C Matabosch, D Fofi, J Forest, A review of recent range image registration methods with accuracy evaluation. *Image Vis. Comput.* **25**, 578–596 (2007)
8. L Silva, ORP Bellon, KL Boyer, Multiview range image registration using the surface interpenetration measure. *Image Vis. Comput.* **25**(1), 114–125 (2007)
9. T Masuda, Log-polar height maps for multiple range image registration. *Comput. Vis. Image Understand.* **113**(11), 1158–1169 (2009)
10. PJ Besl, ND McKay, A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(2), 239–256 (1992)
11. S Rusinkiewicz, M Levoy, in *3rd International Conference on 3D Digital Imaging and Modeling (3DIM 2001)*. Efficient variants of the ICP algorithm (IEEE Computer Society, Quebec City, Canada, 1 June 2001 28 May), pp. 145–152. <http://csdl.computer.org/comp/proceedings/3dim/2001/0984/00/09840145abs.htm>
12. MA Fischler, RC Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *ACM Commun.* **24**(6), 381–395 (1981)
13. CS Chua, R Jarvis, 3D free-form surface registration and object recognition. *Int. J. Comput. Vis.* **17**(1), 77–99 (1996)
14. D Aiger, NJ Mitra, D Cohen-Or, in *ACM SIGGRAPH 2008 papers*. 4-points congruent sets for robust pairwise surface registration (ACM, New York, NY, USA, 2008), pp. 85:1–85:10
15. AM Bronstein, MM Bronstein, M Ovsjanikov, in *3D features, surface descriptors, and object descriptors*, ed. by Y Liu, N Pears. 3D Imaging, Analysis, and Applications (Springer, 2012(to appear))
16. F Stein, G Medioni, Structural indexing: efficient 3-d object recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(2), 125–145 (1992)
17. B Bustos, DA Keim, D Saupe, T Schreck, DV Vranić, Feature-based similarity search in 3D object databases. *ACM Comput. Surv.* **37**, 345–387 (2005)
18. R Toldo, U Castellani, A Fusiello, The bag of words approach for retrieval and categorization of 3D objects. *Vis. Comput.* **26**, 1257–1268 (2010)
19. X Li, I Guskov, in *Proceedings of the third Eurographics symposium on Geometry processing*. Multi-scale features for approximate alignment of point-based surfaces (Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2005). <http://dl.acm.org/citation.cfm?id=1281920.1281955>
20. CH Lee, A Varshney, DW Jacobs, in *ACM SIGGRAPH*. Mesh saliency (ACM, New York, NY, USA, 2005), pp. 659–666. <http://doi.acm.org/10.1145/1073204.1073244>doi:10.1145/1073204.1073244
21. D Lowe, Distinctive image features form scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
22. U Castellani, M Cristani, S Fantoni, V Murino, Sparse points matching by combining 3D mesh saliency with statistical descriptors. *Comp. Graph. Forum.* **27**(2), 643–652 (2008)
23. LJ Skelly, S Sclaroff, in *Proc. SPIE 6762, Two- and Three-Dimensional Methods for Inspection and Metrology V* Improved feature descriptors for 3-d surface matching (vol. SPIE 6762, October 10, 2007), doi:10.1117/12.753263
24. D Thomas, A Sugimoto, Robustly registering range images using local distribution of albedo. *Comput. Vis. Image Understand.* **115**(5), 649–667 (2011)
25. S Rusinkiewicz, O Hall-Holt, M Levoy, Real-time 3D model acquisition. *ACM Trans. Graph.* **21**, 438–446 (2002)
26. S-Y Park, J Baek, J Moon, Hand-held 3D scanning based on coarse and fine registration of multiple range images. *Mach. Vis. Appl.* **22**, 563–579 (2011)
27. T Weise, T Wismer, B Leibe, L Van Gool, Online loop closure for real-time interactive 3D scanning. *Comput. Vis. Image Understand.* **115**, 635–648 (2011)
28. S Izadi, D Kim, O Hilliges, D Molyneaux, R Newcombe, P Kohli, J Shotton, S Hodges, D Freeman, A Davison, A Fitzgibbon, in *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST'11)* Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera (ACM, New York, NY, USA, 2011). <http://doi.acm.org/10.1145/2047196.2047270>doi:10.1145/2047196.2047270
29. BKP Horn, Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am.* **4**, 629–642 (1987)
30. A Johnson, *Spin-Images: A Representation For 3-D Surface Matching*, PhD thesis. ((Carnegie Mellon University, USA, 1997)
31. O Chum, J Matas, in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*. Matching with PROSAC - progressive sample consensus (IEEE Computer Society, Washington, DC, USA, 2005), pp. 220–226. <http://dx.doi.org/10.1109/CVPR.2005.221>doi:10.1109/CVPR.2005.221
32. F Bonarrigo, A Signoroni, in *Proceedings of the 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*. An enhanced Optimization-on-a-Manifold framework for global registration of 3D range data (IEEE Computer Society, Washington, DC, USA, 2011), pp. 350–357. <http://dx.doi.org/10.1109/3DIMPVT.2011.51>doi:10.1109/3DIMPVT.2011.51
33. S Krishnan, PY Lee, JB Moore, S Venkatasubramanian, in *International Journal of Intelligent Systems Technologies and Applications*, vol. 3. Optimisation-on-a-manifold for global registration of multiple 3D point sets (Inderscience Publishers, Inderscience Publishers, Geneva, SWITZERLAND, 2007), pp. 319–340. <http://dx.doi.org/10.1504/IJISTA.2007.014267>doi:10.1504/IJISTA.2007.014267
34. M Kazhdan, M Bolitho, H Hoppe, in *Proceedings of the fourth Eurographics symposium on Geometry processing*. Poisson surface reconstruction (Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2006), pp. 61–70. <http://dl.acm.org/citation.cfm?id=1281957.1281965>
35. A Johnson, M Hebert, Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(5), 433–449 (1999)
36. RI Hartley, A Zisserman, *Multiple View Geometry in Computer Vision*, 2nd edn. (Cambridge University Press, New York, NY, USA, 2003). ISBN: 0521540518

doi:10.1186/1687-6180-2012-148

Cite this article as: Bonarrigo et al.: Multi-view alignment with database of features for an improved usage of high-end 3D scanners. *EURASIP Journal on Advances in Signal Processing* 2012 **2012**:148.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com