ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

Pairwise Registration by Local Orientation Cues

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
Petrelli, A., Di Stefano, L. (2016). Pairwise Registration by Local Orientation Cues. COMPUTER GRAPHICS FORUM, 35(6), 59-72 [10.1111/cgf.12732].

Availability:
This version is available at: https://hdl.handle.net/11585/545388 since: 2016-06-30

Published:
DOI: http://doi.org/10.1111/cgf.12732

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

12 October 2024

This is the final peer-reviewed accepted manuscript of:

Petrelli, A. and Di Stefano, L. (2016), Pairwise Registration by Local Orientation Cues. Computer Graphics Forum, 35: 59-72.

The final published version is available online at: **https://doi.org/10.1111/cgf.12732**

Rights / License:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/)

When citing, please refer to the published version.

Volume xx (200y), Number z, pp. 1–15

Pairwise registration by local orientation cues

Alioscia Petrelli and Luigi Di Stefano

Department of Computer Science and Engineering, University of Bologna, Bologna, Italy
{alioscia.petrelli, luigi.distefano}@unibo.it

Abstract

Inspired by recent work on robust and fast computation of 3D Local Reference Frames (LRF), we propose a novel pipeline for coarse registration of 3D point clouds. Key to the method are: (a) the observation that any two corresponding points endowed with a LRF provide a hypothesis on the rigid motion between two views, (b) the intuition that feature points can be matched based solely on cues directly derived from the computation of the LRF, (c) a feature detection approach relying on a saliency criterion which captures the ability to establish a LRF repeatably. Unlike related work in literature, we also propose a comprehensive experimental evaluation based on diverse kinds of data (such as those acquired by laser scanners, Kinect and stereo cameras) as well as on quantitative comparison with respect to other methods. We also address the issue of setting the many parameters that characterize coarse registration pipelines fairly and realistically. The experimental evaluation vouches that our method can handle effectively data acquired by different sensors and is remarkably fast.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems I.4.3 [Image Processing and Computer Vision]: Enhancement—Registration I.4.6 [Image Processing and Computer Vision]: Segmentation—Edge and feature detection

1. Introduction

The majority of 3D sensors used to acquire shape information, such as laser scanners, stereo and time-of-flight cameras, structured light systems, share the inability to scan the entire object at once. Indeed, a single acquisition captures only the portion of the object seen from the viewpoint of the sensor. This issue can be dealt with by acquiring the object from different vantage points so to cover the entire surface. Yet, upon completion of the acquisition session, the partial views (hereinafter also referred to as scans) are not aligned coherently but each is represented in its own reference frame as determined by the vantage point of the sensor. Therefore, views have to be aligned with respect to a unique reference to obtain the final reconstruction of the object. This process, known as *3D Registration*, is most desirably accomplished without assumptions on the relative position and orientation of the views at hand and it is typically addressed by trying to determine the rigid motion that aligns any of the two views at a time. This alignment task, referred to as *Pairwise Registration*, is ordinarily faced by a two-step procedure. The aim of the former is solely to provide a sufficiently correct alignment to the latter, which then attends to refine the registration until it converges. On condition of such good initial guess, the second step is solved effectively by the ubiquitous *Iterative closest points* algorithm (ICP) [YM92, BM92], or one of its variants [RL01, LW09]. While research on fine registration can be considered to be quite consolidated, new approaches to the open problem of the initial coarse alignment are proposed every year in literature.

The coarse registration methods published during the last two decades can be categorized into global and local approaches. Prominent methods within the former category rely on *Principal Component Analysis* [CYL98], on *Extended Gaussian Images* [MPD06], as introduced by Horn in [Hor84], as well as on *Algebraic Surface Models* [TCC98]. All these methods try to find the mapping between the two surfaces by computing descriptions that globally represent the views. Such criteria find it difficult to deal with nuisances such as point density variation, noise and missing regions due to self-occlusions, thereby failing when the overlap between the two surfaces is limited. To overcome these issues, local methods have become established. A set of feature points is extracted from both views and the local neighbourhood of each point (hereinafter *support*) is projected onto a

suitable feature space so as to obtain a description invariant (or covariant) to pose and as robust as possible to the nuisances induced by acquisition. Then, correspondences between local features are established based on similarities of descriptions, so that, eventually, the rigid motion that best aligns corresponding feature points is easily computed by a robust estimator, such as e.g. *RANSAC* [FB81]. Local methods have proved to be significantly resilient to nuisances such as those mentioned above and hence allow registering view pairs that share just a limited portion of surface.

Indeed, not all the proposed feature descriptors are inherently pose-invariant. For example, both [BSL11] and [FHK*04] compute multiple descriptions, one for each angular subdivision of the support. This requires the matching stage to perform circular shifts in order to evaluate the similarity between two descriptors. In other proposals, though, the description method is itself endowed with invariance to pose. *Spin Images* [JH99], *FPFH* [RBB09] and *Normal/Integral Hash* [ART10] are histograms wherein the contributions of the points within the support are related to the normal at the feature point only. [CCFM08] treats feature points by *Hidden Markov Models* that intrinsically own pose invariance, whereas [LG05] and [Mas09] take the magnitude of the Fourier Transform of the descriptor, thereby gaining invariance to rotation which is encoded into the phase. However, the most widespread approach ([KK12, MBO06, BNSN12, dSFMMH11, SM92, CJ97, SA01, Zho09, TSD10b, TSD10a, KPW*10, ZBH12]) to attain pose-invariance deploys a Local Reference Frame (LRF) centered on the feature point and attached to the surface regardless of its orientation. Thereby, description can encode local shape traits with respect to a canonical reference associated with the feature point. The findings reported in [TSD10b, PD11, PDS12] highlight clearly how the repeatability† of the computation of the LRF is key to robustness of the description and, accordingly, to the performance of the overall feature matching process.

Although effective and fast algorithms pertaining their computation have been devised [TSD10b, PD11, dSFMMH11, PDS12], in the field of registration LRFs have so far only been considered instrumental to feature description. Conversely, in this paper we propose a different and novel registration paradigm, which stems from the observations that LRFs can indeed provide basic shape cues and that two corresponding points equipped with their LRFs allows the rigid motion that aligns two views to be computed. More precisely, we rely on the method proposed in [PDS12] to compute highly repeatable LRFs at feature points and show how such computation provides the core of a coarse registration pipeline which does not require a costly feature description stage. Thanks to the minimal feature description, which also implies a light downstream correspondence process, the ensuing pipeline turns out remarkably fast without any loss of registration efficacy.

It is worth pointing out that the idea of using the LRF attached to points to estimate the rototranslation to align two views can be found also in [MW08], [FA96] and [MBO06]. However, such papers rely on a single correspondence, whilst we deploy an *Hough Voting* scheme [TDS10] to robustly account simultaneously for many correspondences. More importantly, the pipeline in [MBO06] follows the standard paradigm whereby LRFs are primarily deployed to establish a canonical reference for the purpose of feature description. On the other hand, the work in [FA96] turns out infeasible unless views consist of quite a small number of points, as it consists in a RANSAC-based approach bound to operate with just a small fraction of outliers and it mandates running as many nearest neighbour searches as the number of points to determine the consensus set.

Although our registration pipeline can be feed with any kind of 3D feature points, as a second contribution of this paper we propose a novel detector specifically conceived to provide features suited to our method. In particular, we argue that the underling saliency cue should capture the *orientability* of features, i.e. the ability to compute the LRF repeatably despite feature localization being possibly inaccurate. Accordingly, we develop an efficient algorithm which conveniently deploys the observed relationship between *orientability* and *flatness* to quickly extract features particularly suitable for our pipeline and uniformly distributed throughout the views, the latter being a beneficial property for coarse registration.

The third contribution of this paper concerns the experimental evaluation. Indeed, between all those cited up to this point, most papers present experiments on data acquired with a sole type of sensor, while just a few of them consider more than one modality. Even more questionably, no paper attempts a comparison to other proposals. Usually, only the assessment between different variants of the proposed algorithm is presented and, sometimes, no quantitative results are provided. The only exception concerns [MBO06], which considers three well-known datasets and compares the proposed method to a pipeline based on *Spin Images*. Furthermore, though a significant number of surveys have been published (see e.g. [TCL*13]), only one experimental evaluation has been issued ([SMFF07]). Unfortunately, it considers proposals that nowadays would be regarded as baseline methods. Such condition of things might be due to the lack of an acknowledged benchmark, let alone standard methodologies, on which to ground the evaluation process. This issue is worsened by the need to obtain and make proper use of the author's original implementation for the sake of fairness

† An LRF algorithm is said to be repeatable when it provides the same canonical orientation across different views of a surface patch. Specific figures of merit to quantify this property have been proposed in literature [TSD10b, PD11, PDS12].

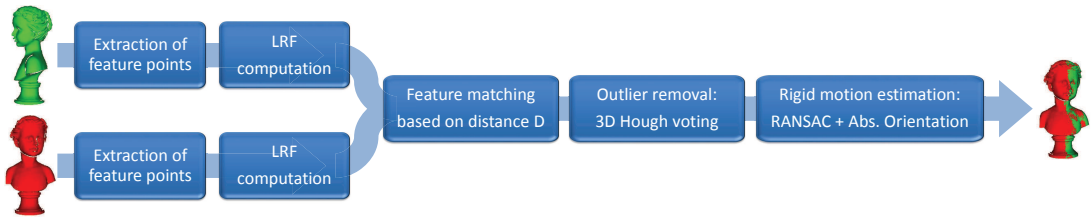

Figure 1: *Outline of the proposed pipeline: LRF computation is key to match feature points based on an elementary shape cue (D) as well as to prune most outliers by Hough Voting. The final pose estimation is accomplished by RANSAC.*

in the evaluation, as well as by that of running lengthy processes to tune, on diverse kinds of data, the many parameters which typically characterize coarse registration algorithms. As a result, we found it exceedingly difficult, if not impossible, to shed light on which coarse registration pipelines are most effective and under which conditions.

In this paper we, therefore, carry out an extensive experimental evaluation on a large ensemble of datasets acquired with different sensors and show how our method easily adapts to the different modalities. Moreover, we compare quantitatively our proposal to a pipeline relying on the popular *Spin Image* descriptor as well as to two more recent methods. The comparison neatly proving that our algorithm is capable of aligning many views that cannot be handled by the three other considered methods. Regarding the *Spin Image* pipeline (SI), proposed in [JH97], we have used the implementation available in the *Mesh Toolbox*[‡]. Concerning more recent methods, the former is the *4-Points Congruent Set* algorithm (4PCS) by Aiger et al. [AMCO08] while the latter is a recent local approach introduced in [BSL11, BSL12] (BSL12) which works on range images only.

The paper is structured as follows. Section 2 describes the proposed coarse registration pipeline based on coherently aligning the LRFs attached to a set of feature points extracted from two views. Section 3 addressed the design of a feature extraction algorithm conceived to detect points suited to improve the effectiveness of the proposed pipeline without penalizing its computational efficiency. The next section concerns the experimental evaluation: we list the considered datasets, explain the adopted methodology and provide both quantitative as well as qualitative results. Section 5 concludes the paper by outlining some planned extensions of this work.

2. Pipeline Description

Given two partial views of an object acquired from different vantage points, V_I and V_J, the aim of a pairwise registration pipeline is to find the rigid motion that aligns the views so that the shared surface portions do overlap. Our method, outlined in Fig. 1, starts by extracting two sets of feature points, F_I and F_J, from the two views. This can be achieved either by random selection of a given number of points or by the feature detection algorithm described in Section 3.

The second step of our pipeline differs significantly from mainstream literature approaches. As explained in Section 1, we dismiss the time-consuming description stage carried out at this level of the pipeline and instead keep only the Local Reference Frame (LRF) computation ordinarily devoted to endowing description with invariance to pose. Purposely, we deploy the method introduced in [PDS12], which requires the normals[§] associated to points and the computation of the *mesh resolution* (hereinafter *mr*, computed as either the average length of the edges of the meshes or, should the dataset consist of point clouds, the average distance between neighbouring points). Given a feature point p, the algorithm starts by robustly estimating the z axis (colored in blue in Fig. 2) as the normal to the plane, π_z, that best fits the points within a small spherical support of radius R_z centered at p (depicted in blue in Fig.2). The sign of the z axis is disambiguated so as to orient it coherently to the average normal computed over support points. It is worth pointing out that the algorithm is not critically affected by the stability of the computed normals as they are used - after having been averaged - only to disambiguate the sign of the z axis. A different support is considered for estimation of the x axis, namely the intersection between the surface and the spherical shell centered at the feature point and defined by the radii pair $[0.85 \times R_x, R_x]$ (depicted in red points in Fig.2). Considering such points, the signed distance from plane π_z is evaluated and the point, p_D, exhibiting the largest distance, D, is selected. The x axis (represented in red in Fig. 2), is attained by projecting onto π_z the vector from the feature point to p_D. The y unit vector (shown in green in Fig. 2) is then given by the cross-product $z \times x$.

The method in [PDS12] possesses traits that render it par-

[‡] http://www.cs.cmu.edu/~vmr/software/meshtoolbox/introduction.html

[§] The normals are computed by the *vtkPolyDataNormals* function of the VTK library, which computes the normal of each triangle and, then, computes the normal of each point by averaging the normals of the triangles connected to the point.

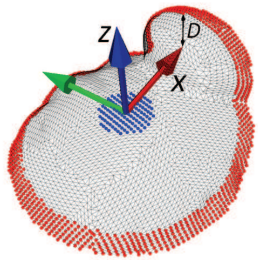

Figure 2: *An example helping to describe the computation of the local reference frames.*

ticularly suited to the task of registration. First of all, the repeatability of the LRFs tends to increase significantly with the support size R_x. The main nuisance that would hinder repeatability along with increasing such a size turning out clutter, which, however, is not present in the registration task. This vouches that the method is robust to missing surface portions within the neighbourhood of a point, as it would happen at features located close to the boundaries of a view. Indeed, the repeatability of a LRF only depends on whether point p_D ends up or not in a missing region, such "highest" points tending to better withstand changes of the vantage point as they are less likely to be occluded by other surface patches. This is a quite favorable property in registration applications, as, when the views in a pair are acquired from angularly distant viewpoints, and thus are hard to align, most of the limited overlap is found at boundary regions. Another benefit of the approach dwells on its robustness to point density variations, both uniform, as induced by changes of the acquisition distance, and non-uniform, as determined by out-of-plane rotations of the sensor. Finally, the algorithm comes out fast even when applied to wide supports due to the small fraction of points actually involved in the computation.

Any pair of corresponding feature points equipped with correctly established LRFs defines the sought rigid motion. Stemming from this observation, the next stages of our pipeline aim at sifting out from the set of all the $F_I \times F_J$ candidate pairs a sizable subset of correspondences to estimate the rototranslation to align the views, i.e. to bring them to the same reference frame. We found a good cue to be the distance, D, inherently associated with each feature point upon computation of the LRF. Indeed, as the LRF algorithm establishes a canonical reference, D turns out to be a basic measurement related to the local shape of the surface around the feature point. Accordingly, assuming LRFs to have been correctly established, corresponding features should exhibit similar D values. This property can be exploited to discard candidate correspondences between feature points showing significantly different D values. To this purpose, the difference $D_{ij} = \left|D_i - D_j\right|$ is computed and normalized with respect to $D_{max} = \max_{i,j}\left(D_{ij}\right)$ for each candidate pair. If the normalized difference, $\widetilde{D}_{ij} = \frac{D_{ij}}{D_{max}}$, is above a threshold, T_D, the pair is discarded. This is vouched by Fig. 4, in which we consider two very different datasets and show as a blue curve the *pdf* of $\widetilde{D}_{ij}$, $p\left(\widetilde{D}_{ij}|P\right)$, for good correspondences, P. Despite the diversity in the data, both curves look very similar and clearly show that when $\widetilde{D}_{ij}$ exceeds a certain threshold (such as e.g. 0.2) a feature correspondence is unlikely to be correct.

The proposed matching process is really fast, as it boils down to sorting both the features in F_I and F_J with respect to D and simultaneously scanning the sorted lists to collect those pairs that do not satisfy the previous pruning condition. In standard pipelines, instead, the matching stage is typically expensive as it involves computing distances in a high-dimensional description space so cyanas to retain pairs of features appearing close one to another in that space. Furthermore, if either of the two supports around corresponding features gets spoiled due to missing surface regions, the associated descriptors will be corrupted alike, so that, typically, the features would result far away one to another in the description space. This does not happen in our matching scheme due to the LRF algorithm being resilient to missing regions: as long as the LRFs have been correctly established, the D values related to two corresponding features turn out similar. The proposed matching approach turns out also very robust to noise. Indeed, the D value of the "highest" point p_D is typically large and, as such, it is unlikely that noise at p_D would decrease D as much as to promote another point far away from p_D to become the "highest" point. More likely, noise may promote another point in the vicinity of p_D, which, however, implies a tolerable error. Fig. 3 shows the result of an experiment carried out on the *Bunny* dataset in order to analyze the behavior of the matcher while different levels of Gaussian noise are injected into the data. For each view, 2000 points have been randomly extracted and matched setting $T_D = 0.01$ (see Table 1) so to evaluate the mean number of correct (TP) and wrong (FP) correspondences established across all 45 view pairs. Moreover, the chart reports the number of correctly aligned LRFs (TP_LRF) according to the figure of merit $\bar{A}$ introduced in [PDS12]. Hence, Fig. 3 highlights clearly the robustness to noise of the matching process as the curves remain very stable as the noise level increases.

On the other hand, our basic shape cue D has a rather poor discriminative power compared to a high-dimensional descriptor, so that measuring close (or even identical) values cannot be treated as a sufficient condition to declare two features as corresponding. This is shown, again, in Fig. 4, where the red curves represent $p\left(\widetilde{D}_{ij}|N\right)$, i.e. the *pdf* of $\widetilde{D}_{ij}$ concerning wrong correspondences (N). Though choosing a suitable threshold, T_D, somehow in the range $[0, 0.2]$, assures the preservation of the majority of inliers, it still brings in a large percentage of false correspondences. Therefore, we further prune outliers by enforcing geometric consistency constraints according to the *Hough Voting* method

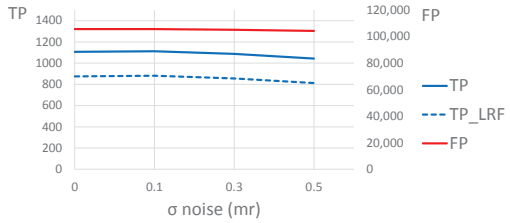

Figure 3: *Feature matching on the Bunny dataset with increasing noise. The blue and red solid curves report, respectively, the average number of correct (left vertical axis) and wrong (right vertical axis) correspondences. The blue dashed curve highlights the average number of correct correspondences yielding aligned LRFs.*

proposed in [TDS10], which, again, relies on the availability of repeatable LRFs attached to features. First, for each feature F_j in V_J, the centroid C_J of V_J is expressed with respect to LRF_j, i.e. the canonical reference attached to F_j, in order to obtain the set of $\mathcal{LRF}_J^j(C_J)$, where the notation $\mathcal{LRF}_J^j(.)$ expresses the change of basis from the global reference frame of V_J to LRF_j. Then, for each pair of correspondences (F_i, F_j), the change of basis from LRF_i to the global reference of V_I, $\mathcal{LRF}_i^I(.)$ is computed and applied to the transformed centroid of V_J: $\mathcal{LRF}_i^I\left(\mathcal{LRF}_J^j(C_J)\right)$. In other words, for each pair of correspondences, a candidate roto-translation $\mathcal{RT}_j^i(.)$ is estimated as the transformation that aligns LRF_j to LRF_i: $\mathcal{RT}_j^i(.) = \mathcal{LRF}_i^I\left(\mathcal{LRF}_J^j(.)\right)$. This is used to move the centroid C_J of V_J to the new position $\mathcal{RT}_j^i(C_J)$. Correct correspondences (F_i, F_j) will vote coherently for the position of C_J in V_I, i.e. the estimated motions will tend to localize $\mathcal{RT}_j^i(C_J)$ in a unique position within the 3D volume associated with V_I, which is referred to as Hough Space in [TDS10].

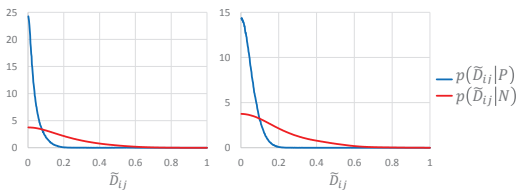

Figure 4: *Probability density functions of normalized differences, $\widetilde{D}_{ij}$, for feature correspondences dealing with the high-quality Neptune dataset (left) and low-quality Duck-Kinect (right) dataset (see sec. 4.1). Blue and red curves concern correct and wrong correspondences respectively.*

To implement the Hough Space, we compute the centroid and standard deviations of the x, y, z coordinates of the points in V_I. The origin of the Hough Space is given by the centroid and each dimension is taken as large as 4 times the corresponding standard deviation, so as to consider about 95% of the points of V_I and neglect possible outliers far away from the centroid. Moreover, the size of each dimension is further enlarged by a factor f_{hough}, to allow rotated centroids to fall outside the tight bounding volume around V_I. The thus defined volume is evenly quantized into cubic bins of side S_{bin}. Each of the $\mathcal{RT}_j^i(C_J)$ votes for the bin hit by the roto-translated centroid. Then, bin scores are computed by accumulating the votes falling into the $3 \times 3 \times 3$ neighbourhood centered at each bin. Eventually, the bin showing the highest score is selected in order to sift out the correspondences to be used to estimate the rigid motion to align the views.

The matching process based on the distance D and the Hough voting stage contribute synergistically to effective filtering of wrong correspondences. For example, considering again the experiment of Fig. 3, when exacting 2000 random features from each partial view of *Bunny*, the matching stage sifts out, on average, 2.66% of all the $F_I \times F_J$ candidate pairs, i.e. 106768 correspondences, whereas the further pruning performed by the Hough voting delivers about 0.54% of the pairs provided by the matching process, so to forward to the final stage of our pipeline only 561 correspondences on average.

Nonetheless, such pairs are not guaranteed to be inlier correspondences only. Indeed, for the sake of memory efficiency, the method in [TDS10] relies on a 3-dimensional (translation only) rather than 6-dimensional (translation plus rotation) Hough Space, thereby allowing, in principle, different rigid motion hypotheses to vote for the same bin. Likewise, quantization effects may determine different hypotheses to collapse into a single bin. Therefore, given the correspondences associated with both the highest bin and its neighboring bins, we carry out the final rigid motion estimation stage robustly by applying the standard Absolute Orientation algorithm proposed in [Hor87] within the RANSAC framework.

3. Feature Extraction

The registration pipeline described so far is agnostic with respect to the kind of features extracted in the first stage. As such, one may rely on random extraction of feature points or use any of the several 3D detectors proposed in literature (see [STD13] for a recent survey and evaluation of prominent proposals).

Unfortunately, the evaluation in [STD13] highlights that the computational efficiency of all considered proposals is by far unsatisfactory, so there is no algorithm that may be plugged into our pipeline without exceedingly slowing down the computation. Moreover, existing detectors rely on maximizing a specific saliency criterion which inherently privileges certain shape structures so that, generally, features tend to cluster in some areas rather than scatter uniformly across

a view. However, for the purpose of accurate estimation of the rigid motion between two views, it is highly beneficial to rely on features as uniformly distributed as possible across the views. Accordingly, we conjecture that a suitable feature detector for coarse registration should better provide a good trade-off between saliency and uniformity rather than maximize saliency.

Based on the above considerations, investigation on suitable features to be fed into our pipeline seems to call for the design of a novel 3D detector that would provide rapidly salient and uniformly distributed points. First of all, this requires reasoning about the saliency criterion. Unlike mainstream work on the subject, our coarse registration pipeline is rooted only on the ability to compute LRFs repeatably despite nuisances. Accordingly, a suitable saliency criterion would capture this ability: "good" features for our pipeline are points where the LRF can be computed repeatably. We dub *orientability* such a peculiar saliency criterion[¶].

In the LRF algorithm proposed in [PDS12], the repeatability of the x axis is significantly dependent on the stability of the z axis, as the latter provides the reference plane to compute the signed distances that then would define the former. Thus, as the stability of the z axis is clearly highest at points located in flat surface areas, it seems that with the method in [PDS12] there is an inherent relationship between *flatness* and *orientability*. To validate this intuition, we performed a qualitative experimental study aimed at comparing the *flatness* and *orientability* of surface points. Given a surface point p together with its normal n_p, the *flatness* at p is higher as the normals at the points within a neighbourhood of p are more closely aligned to n_p. Therefore, we define the *flatness* at p as the mean cosine between n_p and the normals at the points within a sphere of radius R_f centered at p. As the nuisance inherently associated with feature extraction is imprecise localization of feature points, the resilience to be captured by a proper *orientability* notion should address this type of nuisance. Thus, given a point p, the corresponding points p_j in the other views of a dataset are determined by applying *ground truth* rigid motions between the views (ground truth information is available for all the datasets considered in this paper). According to the above mentioned notion, p would exhibit high *orientability* whenever the LRF computed at p is correctly aligned to those computed at points p_j despite localization of the latter turning out imprecise. Hence, to capture this property, we compute the LRF at all the points, $p_{k,j}$ falling within a neighbourhood centered at each of the p_j, so as to then establish whether the LRF computed at p is correctly aligned or not to that computed at each $p_{k,j}$ (i.e. aligned or not to that computed at corresponding though imprecisely localized features). To establish whether any two such LRFs are correctly aligned or not we rely on the repeatability criterion proposed in [PDS12] and, accordingly, calculate the *orientability* index at p as the percentage of correctly aligned LRFs.

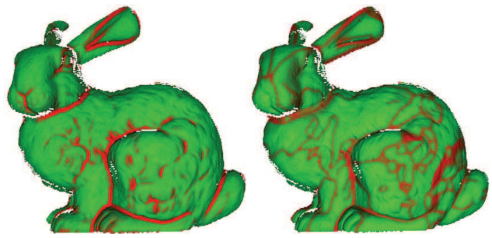

Figure 5: *Flatness Vs Orientability. On the left the flatness map of a view, on the right the corresponding orientability map. The green colour represent flat and highly-orientable points respectively, red points feature high curvature in the flatness map and poor LRF repeatability in the orientability map.*

Fig. 5 allows a visual comparison of the *flatness* and *orientability* maps of a partial view of the *Bunny* object of the Stanford Repository [CL96]. As a first consideration, the left map shows that our measure of flatness captures the curvature of the shape properly. Besides, as for the orientability map, it is worth highlighting the high orientability of the majority of points as a further proof of the effectiveness of the method adopted to compute LRFs. But more importantly, the comparison shows clearly that there exists a relationship between flatness and orientability. In particular, many red points in the left map turn out red also in the right one: the repeatability of the LRFs is usually poor at surface areas featuring a pronounced curvature so that the computation of the z axis turns out to be unstable (e.g. the ears of the *Bunny*).

As in real registration settings the rigid motions between views are obviously not available (we indeed seek to estimate them!), the defined orientability index cannot be measured directly in practice. Hence, the idea stemming from our analysis is to try to extract flat points instead, as with respect to our pipeline they are more likely to be salient (i.e. orientable) than high-curvature ones. As such, whereas most detectors present in literature are based on extraction of points exhibiting high curvature, we take, somehow paradoxically, just the opposite path. This approach provides at least two additional benefits, though. Firstly, computation of flatness is fast as it involves only inner products within a small supporting neighbourhood (as shown in Table 1 a small R_f suffices). Secondly, flat surface areas are less prone than high-curvature ones to self-occlusions caused by out-of-plane rotations of the sensor.

The devised feature detector is described with the aid of Fig. 6. The algorithm repeatedly extracts a random point p_{seed} (shown in light green in the zoom-in panel on the left

¶ Our notion of *orientability* differs from the use of this term to denote the property of consistently disambiguating the sign of the normal at every point of a surface.

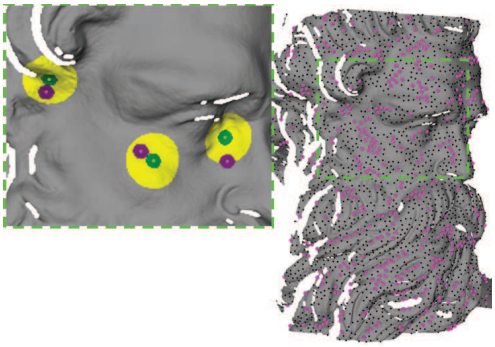

Figure 6: *Exemplar feature extraction by the proposed algorithm. The zoom-in panel on the left shows the detection process at a glance: for each randomly chosen seed point (in green), the flattest point (in fuchsia) in its neighbourhood is extracted. The image on the right highlights the two-step extraction process: the points selected by the first step are shown as black smaller dots, while the final features provided by the second step as larger ones.*

side of the Figure). Then, the flatness index is computed at all the points within a spherical support of radius R_{search} centered at p_{seed} (highlighted in yellow in the panel), so as to pick-up as feature the point, p_{max}, turning out maximally flat (coloured in fuchsia in the panel). To avoid further detections in the proximity of already selected features, the points at a distance lesser than $R_{discard}$ from p_{max} (in purple in the panel) are pruned from the set of candidate feature points, and, alike, those around p_{seed} according to $R_{discard}$ pruned from the set of candidate random seeds (in green in the panel).

The method continues to iterate until the percentage of discarded points, either as potential feature, p_{max}, or random seed, p_{seed}, gets higher than a threshold, T_{area}, which is tightly correlated to the fraction of the view that one wishes to explore during the feature extraction process. As for the requirement to trade-off between saliency and uniformity, random extraction of seeds ensures feature points to be scattered throughout the partial view, while the subsequent flatness maximization weighs in favor of saliency. Regarding efficiency, the critical step of the algorithm consists in gathering the points inside the support of radius R_{search}, which needs to be large enough in size and thus may slow down the adopted kd-tree search. To overcome this issue, we devised the two-step approach illustrated in the right image of Fig. 6. In the first step, the method is applied to the whole view with a small radius, R^1_{search}, to search for maximally flat points around random seeds. Due to R^1_{search} being small, the first step efficiently subsamples the view so as to provide a subset of candidate flat points. The second step consists of running the process with a larger radius, R^2_{search}, on the subsampled view made out of the candidate features provided by the first step only. Accordingly, much fewer instances of the slower search do take place and efficiency is not penalized. The termination threshold for the first step, T^1_{area}, is set high enough to explore a large fraction of the input view, whereas the threshold for the second step, T^2_{area}, represents the parameter that actually controls the amount of extracted features points.

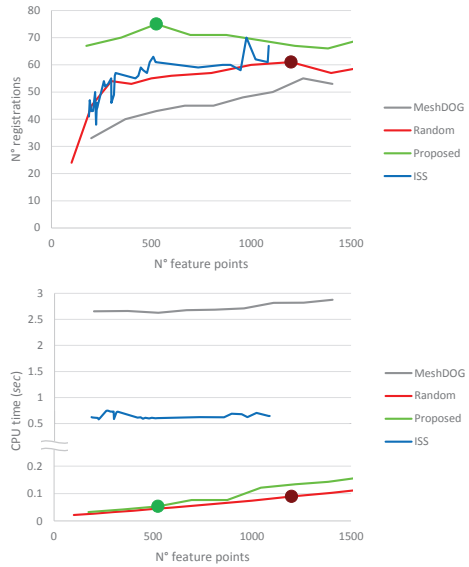

Figure 7: *Comparison on the Buste dataset between proposed detector, random extraction, ISS and MeshDOG. In both charts the figure of merit is plotted as a function of the mean number of extracted feature points. The top chart reports the number of view pairs aligned correctly by our pipeline, whereas the bottom one shows the mean computation time required to align two views. To better compare our proposal to the random detector, the working points providing the highest number of correct alignments are highlighted by dots in both charts. Sec. 4 explains how the figures of merit plotted in the two charts are calculated.*

We performed comprehensive experiments to validate the improvement brought in by the proposed feature extraction algorithm. In particular, we compared the performance delivered by our pipeline with features provided by the proposed detector, random extraction (our initial choice), Intrinsic Shape Signatures (*ISS*) [Zho09] and *MeshDOG* [ZBH12]. According to the evaluation in [STD13], *ISS* and *MeshDOG* may be regarded as prominent fixed-scale and adaptive-scale detectors, respectively. Indeed, in their respective categories, both turn out to be the fastest and rank quite high in terms of repeatability. We found that the pipeline deploying the proposed detector delivers the best performance consistently across datasets: it can align a higher number of view pairs, and, in the event of similar registration rates, it comes out, on the whole, the fastest. This

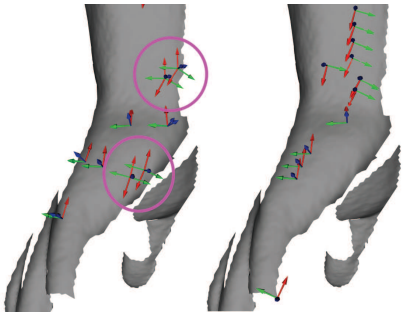

Figure 8: *LRFs computed at corresponding features found on a portion of Neptune's hand. The left image deals with randomly extracted features, the right image with features detected by our algorithm. Both images show the LRFs computed in V_I together with those computed in V_J and transformed into V_I according to the ground truth rigid motion. Misaligned LRFs are highlighted by the pink circles.*

is perhaps surprising, as one might guess that random detection would deliver the highest efficiency. However, though our detection scheme is obviously slower than random extraction, it provides better features, that is, according to our saliency notion, inherently endowed with more repeatable LRFs, so that a smaller amount of features needs to be forwarded to the next stages of the pipeline, which makes the overall computation faster.

The above described behavior is well pictured in Fig. 7, which shows the results regarding the registration of all the view pairs composing the *Buste* dataset (see sec. 4.1). The charts report the number of correctly aligned view pairs (top) and the mean CPU time to align two views (bottom) as a function of the average number of feature points detected in the partial views, which can be controlled by the user through a parameter in each of the four considered detectors. The top chart highlights how, regardless of the chosen number of extracted features, the proposed detector significantly improves the effectiveness of our pipeline with respect to the other detectors. It is worth highlighting here that, although any feature detection algorithm is inherently more repeatable than the random detector, the kind of saliency deployed by *ISS* and *MeshDOG* does not seem particularly suited to our pipeline. Indeed, the improvement provided by ISS over random extraction is modest on average and also not consistent across working points, whereas using the keypoints extracted by *MeshDOG* leads to lower registration rates than randomly extracted features. This may be explained by observing that, as also illustrated in Fig. 2 and Fig. 5 of [STD13], *ISS* and *MeshDOG* tend to fire on high-curvature structures, like those depicted in red in Fig. 5 of this paper, than on flattish areas, as instead would be required by the saliency notion deployed in our pipeline. As for the bottom chart of Fig. 7, at first sight the computational efficiency would seem somehow comparable between our detector and random extraction, with *ISS* and *MeshDOG* definitely turning out to be slower[‖]. However, a deeper analysis reveals that our detector can improve the efficiency of the pipeline. Indeed, as highlighted by the two dots in the top chart, choosing the feature quantity that yields the highest number of correctly aligned pairs for both the random detector and our detector, we end up requiring 1200 features (so to align 61 pairs) and 525 features (so to align 75 pairs) respectively. As pointed out by the dots in the bottom chart, at these two working points the proposed detector does render the pipeline faster than random extraction of features.

Finally, in Fig. 8 we show a qualitative experiment aimed at comparing the repeatability of the LRFs computed at random features and feature points detected by our algorithm. We consider two views, V_I and V_J, of the *Neptune* dataset (see sec. 4.1), extract feature points by both methods and detect correspondences based on the *ground truth* alignment transformation. Then, for the two methods, we compute the LRFs for each pair of corresponding features and, deploying again the ground truth rigid motion, draw both the LRFs in one view (i.e. V_I): the two LRFs drawn at each point will look more aligned as they have been computed more repeatably in the two views. Accordingly, in the right image (our detector) all LRF pairs look correctly aligned, whilst notable misalignments can be perceived in the pairs depicted in the left image (random detector).

4. Experimental Evaluation

In this section we compare the performance of our method, referred to here as LRF, to those yielded by SI [JH97], 4PCS [AMCO08] and BSL12 [BSL11,BSL12]. The original implementations of the three algorithms have been kindly provided by the authors who also helped us a great deal through personal communications to tune the parameters of their methods properly. However, as far as SI is concerned, instead of describing all the points in one view and a subset in the other, as done in the original code, we have introduced a slight modification in order to execute random extraction of keypoints in both views. This has been necessary as otherwise the pipeline would have resulted exceedingly slow, making it impossible to complete most of the experiments. Indeed, even with the introduced modification, it turned out infeasible to follow out the experiments on the datasets comprising the largest number of views, i.e. *OilPump*, *Venus* and *Shell*.

‖ Due to software compatibility issues we could not run *MeshDOG* on the machine used to measure the computation times of the other detectors. Thus, the timings for *MeshDOG* reported in the bottom chart of Fig. 7 represent our best estimation of the actual efficiency of the algorithm.

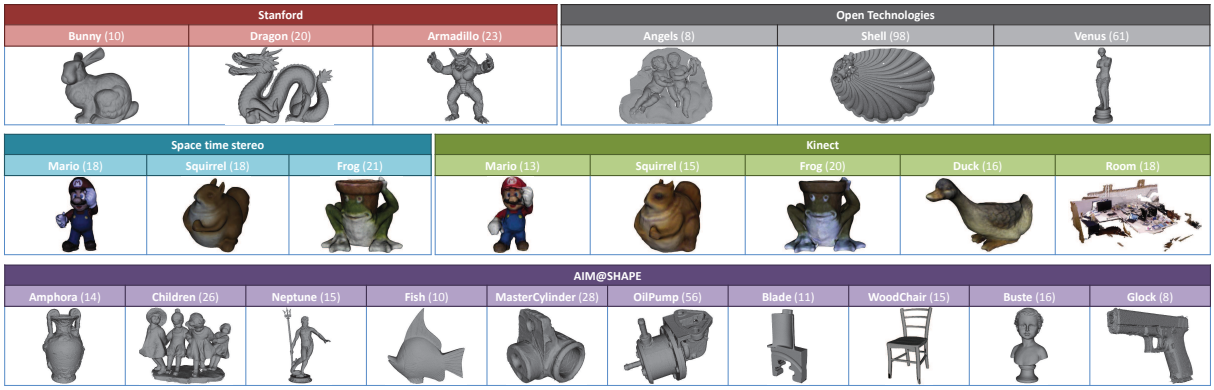

Figure 9: *Thumbnails of the 24 datasets considered in the experimental evaluation. For each dataset the number of views, M, is reported between brackets.*

4.1. Datasets

An essential trait of a registration pipeline is to work successfully with different sensing modalities and under various real working conditions and nuisances. For this reason, as mentioned in the introduction, we evaluate the considered pipelines on the extensive collection of datasets detailed in Figure 9. Each dataset consists of a number of partial views for which normals are computed and the ground truth rigid motion between each view pair is available. Three of the datasets are taken from the very popular Stanford Repository [CL96], namely *Bunny*, *Armadillo* and *Dragon*, all acquired with a Cyberware 3030 MS laser scanner. Many other datasets come from the AIM@SHAPE Repository**, and have been acquired with different scanners such as the very accurate Minolta VI910 (*Amphora*, *Children*, *Neptune*, *Fish*, *MasterCylinder*, *OilPump*, *Blade*, *WoodChair*), the Roland LPX-250 (*Buste*), and the low quality Minolta VI700 (*Glock*). Then, to extend the evaluation to different sensing modalities, we acquired four datasets in our laboratory by means of a Kinect device (*MarioKinect*, *DuckKinect*, *FrogKinect* and *SquirrelKinect*) and three datasets by a Spacetime Stereo set-up [DNRR05, ZCS03] (*MarioStereo*, *SquirrelStereo*, *FrogStereo*). All our datasets were acquired by rotating the objects in the scene whereas the sensors were kept still. Then, we performed a background subtraction so as to hold only the partial views of the object. After that, we performed a coarse registration and, finally, we carried out a fine global registration by applying the *Scanalyze* tool. As a coarse registration algorithm should provide a rototranslation sufficiently correct to let the subsequent fine registration converge, such ground truth is sufficiently precise for validating the performance of the compared methods. We will make the datasets acquired in our Lab and the code of our method available for research purposes through the project web page. Moreover, we considered the *fr1/room* sequence from the *RGB-D SLAM Benchmark* [SEE*12], which deals with the reconstruction of an indoor scene acquired by a Kinect. In particular, we sampled the sequence every 10 frames so as to obtain a dataset comprising 18 partial views, referred to as *Room* in Figure 9. Finally, we added to the ensemble the three datasets scanned by an high-precision *Open Technologies*® system used for the experimental evaluation in [BSL11, BSL12], i.e. *Angels*, *Venus* and *Shell*. The considered datasets feature different noise levels and point densities. *Kinect* datasets, and *Glock* alike, show the worst quality, with very noisy scans especially along the line-of-sight direction; *Open Technologies*® datasets, on the contrary, consist of very detailed and clean scans; *Spacetime Stereo* allows scans to be obtained with good precision and resolution. Moreover, every dataset comprises a different number of views and different degrees of overlap between them. *OilPump*, *Venus* and *Shell* respectively consist of, 56, 61 and 98 scans. On the other hand, *Glock*, *Angels*, *Bunny* and *Fish* include from 8 to 10 views. Even the physical objects they represents are quite different. *Open Technologies*® datasets deal with large objects, which have been scanned in the context of cultural heritage applications. Conversely, Stanford objects are extremely small. Furthermore, together with complex, rich of features objects, such as *Armadillo*, *Dragon* and *Mario*, we considered a set of mechanical parts (*MasterCylinder*, *OilPump*, *Blade*) as well as the *WoodChair* and *Room* datasets, so as to also address simple shapes and large flat surfaces that, as such, often turn out to be hard to reconstruct due to the scarcity of features. As range images are available only for the Stanford, *Open Technologies*®, Kinect and *MarioStereo* datasets, we have tested BSL12 only on this subset of datasets.

** http://shapes.aim-at-shape.net

LRF					
R_f	$R_{discard}$	R^1_{search}	R^2_{search}	T^1_{search}	T^2_{search}
$5 \cdot mr$	$2 \cdot mr$	$2 \cdot mr$	$20 \cdot mr$	0.9	0.9
R_z		R_x		T_D	
$5 \cdot mr$		$[10, \quad 250] \cdot mr$		0.01	
f_{hough}	S_{bin}	T_{RANSAC}	N_{RANSAC}	P_{RANSAC}	
1.4	$2 \cdot mr$	$8 \cdot mr$	1000	0.99	

SI					
N_{points}	S_{bin}		N_{bin}	T_{angle}	
$[1000, \quad 4000]$	$1, \quad 2, \quad 3 \cdot mr$		20	$\pi/2$	
λ	$T_{saliency}$	T_{corr}	T_{vote}	T_{gc}	T_{ICP}
3	3	0.5	0.5	0.1	$4.0 \cdot mr$

4PCS				
δ	f	T	N_{points}	D_{norm}
$[0.1, \quad 0.5]$	0.7	1.0	$300, \quad 500, \quad 700$	30.0

BSL12			
σ	$N_{octaves}$	N_{maps}	$f_{sampling}$
$[0.5, \quad 4.0]$	3	2	2
M	L	Q	U
3	$12, \quad 24, \quad 36$	150	150

Generalized ICP		
N	ε	R
10	0.1	$8 \cdot mr$

Table 1: *Parameters for the four methods and Generalized ICP.*

LRF			
R_f	$R_{discard}$	R^1_{search}	T^2_{search}
$8 \cdot mr$	$8 \cdot mr$	$8 \cdot mr$	0.5
R_x	S_{bin}	T_{RANSAC}	
$150 \cdot mr$	$6 \cdot mr$	$7 \cdot mr$	

SI		
N_{points}	S_{bin}	T_{ICP}
5000	$4 \cdot mr$	$7 \cdot mr$

4PCS		
δ	f	N_{points}
0.2	0.5	700

BSL12	
σ	L
4.0	36

Table 2: *Different values for Open Technologies® datasets.*

4.2. Methodology

The goal of a coarse registration algorithm is, in practice, to provide an alignment sufficiently correct to then permit a successful fine registration by ICP. Furthermore, even if the task is not subject to real-time constraints, execution time becomes relevant, especially when dealing with datasets comprising large sets of partial views. Finally, for those view pairs that have been coarsely registered, the accuracy of the alignment can be taken as an additional index of the quality of the algorithm. Given a dataset made out of M views, we consider all the $N = \frac{M(M-1)}{2}$ possible view pairs $\{V_I, V_J\}$, and, for each of them, attempt to estimate the rigid motion $\mathcal{RT}(V_J)$ that aligns V_J to V_I by means of the coarse registration algorithm under evaluation. Then, *Generalized ICP* [SHT09] is applied to the pair $\{V_I, \mathcal{RT}(V_J)\}$, and the resulting view $\mathcal{ICP}(V_J)$ is compared to $\mathcal{GT}(V_J)$, the latter being the view obtained by transforming V_J according to the known ground truth rigid motion which aligns V_J to V_I. In particular, if the Root Mean Square Error (RMSE) between $\mathcal{ICP}(V_J)$ and $\mathcal{GT}(V_J)$ is lower than $5 \times mr$, V_I and V_J are judged as correctly registered by the algorithm under evaluation, otherwise a registration failure is recorded. In the event of successful registration, the RMSE between $\mathcal{RT}(V_J)$ and $\mathcal{GT}(V_J)$ is also recorded for the purpose of estimating the accuracy of the algorithm. Therefore, for each dataset and algorithm we collect the following three measurements. N° *Registrations*, i.e. the number of correctly aligned view pairs; *CPU time*, i.e. the average execution time to compute the rigid motion to align a view pair (regardless the outcome being either success or failure); *RMSE*, which represent the average accuracy (i.e. RMSE) across all correctly registered view pairs. It is important to point out that N° *Registrations* is the key performance index that captures the ability of the algorithm to handle view pairs featuring different degrees of overlap. The higher is the registration rate, the more effective in aligning views sharing a limited surface area is the algorithm.

4.3. Parameters

As outlined in sec. 1, coarse registration algorithms include many parameters which are hard to set properly. Although default settings are typically suggested by the authors, more often than not these guidelines come from insights gained by running the algorithm on one specific kind of data. Unfortunately, it turns out that such "standard" values are unlikely to make the method equally effective on diverse data. Thus, running our evaluation using a fixed set of parameters for each method would be unfair and also useless for shedding light on the real limits and merits of the algorithms. On the other hand, to sift out the best from a method in real working conditions, i.e. new data and no ground truth, one should manually set many parameters on a trial-and-error basis, which is simply infeasible. Therefore, we believe that a method capable of working seamlessly on diverse kinds of data should allow the user to get the desired result by setting just one or two parameters by trial-and-error.

Based on these considerations we contacted the authors of the algorithms involved in the evaluation and, following their indications, defined proper default values for all the parameters but the two identified as those most critically affecting performance, which we then tuned optimally on each dataset by an exhaustive grid search so as to maximize the number of correct registrations, N° *Registrations*. In particular, the

two parameters selected by authors of BSL12 are the Gaussian kernel size, σ, to be varied in the range $\{0.5, 4.0\}$ with 0.25 as step, and the number, L, of angular subdivisions of the descriptor, with possible values 12, 24 and 36. As for 4PCS, their authors decided to span δ in the range $\{0.1, 0.5\}$ with increment 0.1 and three possible numbers of extracted points N_{points} (300, 500, 700). Finally, SI has been tested by spanning the number of extracted keypoints, N_{points}, in the range $\{1000, 4000\}$ with step 500 and trying three values (1, 2, 3) for the size of the Spin Image bins. Likewise, we chose suitable defaults for all the parameters of our method but one, i.e. R_x, which was left free to vary in the range $\{10, 250\}$ with step 10 so to maximize N° *Registrations* on each dataset. R_x depends on the mesh resolution as well as on the extent of the flat areas of the objects, and it turns out to be the only parameter of our method that varies significantly across diverse data. Instead, the tuning we performed on the other methods has shown that at least two parameters affect their performance critically. Table 1 summarizes the parameter values adopted for each method, together with those related to *Generalized ICP*.

Open Technologies® datasets are indeed very different from all the other datasets considered in our evaluation, and in particular are characterized by a far higher point density and size. Moreover, BSL12 has been specifically designed to deal with this kind of data and therefore the authors already tuned their parameters optimally for *Open Technologies*® datasets. On the other hand, we found that the default settings chosen for LRF, 4PCS and SI are less appropriate on these datasets. Therefore, for the purpose of comparative evaluation, we found it fairer to also allow the other three methods to determine their default settings for these so diverse data. The authors of 4PCS kindly determined the parameter values of their method, and we did so for LRF and SI. The parameter values that turned out different from Table 1 are listed in Table 2.

4.4. Results

The results of the evaluation are summarized in Table 3, with the adopted color code (the darker the better for all the three indexes) helping to catch, at a glance, the relative performance of the algorithms. Accordingly, our proposal can align a larger number of view pairs with almost all the datasets (21 out of 24). Moreover, in terms of accuracy, our algorithm is only equaled by SI, despite the higher registration rate implying considering more challenging pairs in the computation of the *RMSE* index. As for computational efficiency, BSL12 proves to be the fastest method although our pipeline fairly competes, with 4PCS and SI on the other hand turning out to be notably slower.

A more in-depth analysis of the results highlights that 4PCS overtakes our proposal solely on *MarioKinect* and for a couple of view pairs. Also, 4PCS obtains results comparable to ours on *SquirrelKinect*, *MarioStereo* and *Glock*, and gets good performances, in general, on *Kinect* and *Spacetime Stereo* datasets. However, 4PCS seems less suited to high resolution datasets such as those by *Open Technologies*®. Nonetheless, it is important to recall that, for each dataset, our evaluation provides the result that maximizes the registration rate, regardless of the associated computation time. By comparing the *CPU time* of LRF and 4PCS on *MarioKinect*, *SquirrelKinect*, *MarioStereo* and *Glock*, it is evident that 4PCS spends too much computational effort to obtain these high registration rates. Had the tuning process taken into account constraints on practically acceptable execution times, the registration rates of 4PCS would have turned out notably lower. It is worth observing that, between all datasets, *Neptune* is the one featuring the highest differences of point density between the views, as these include both close-ups on the head and wider scans around the body. The comparison between the performance of 4PCS and LRF on *Neptune* proves that significant benefits can be achieved by the latter, which is designed to handle point density variations robustly. As for BSL12, it turns out to be the best method on high resolution and clean data such as *Open Technologies*® datasets, i.e. the kind of data for which the method has been designed, tuned and tested. *Kinect* data lies exactly on the opposite side: low resolution and very noisy. Interestingly, Table 3 shows BSL12 to be much less effective on such diverse kind of data. It is also worth pointing out that on the *Venus* dataset LRF obtains a registration rate higher than BSL12. The peculiar cylindrical shape of the object causes acquisitions which involve mainly out-of-plane rotations of the sensor. On the contrary, the *Angels* object, a bas-relief, and *Shell*, acquired on one side only, permit a higher number of simpler in-plane rotations of the acquisition system. BSL12 relies on matching feature descriptors computed on 2D supports defined on range images: such kind of supports capture different portions of the physical space around a feature point due to out-of-plane rotations of the vantage point, which inevitably renders feature matching less effective. Differently, LRF relies on 3D supports, which are inherently invariant to any rigid motion of the sensor. Finally, SI provides registration rates similar to 4PCS on the low-precision *Kinect* datasets, and it seems to possess the ability to achieve good performance on the accurate *Open Technologies*® datasets, although this observation relies on the results pertaining the *Angels* and *Venus* datasets only. Generally speaking, even though SI does not turn out to be the best method on any dataset, it demonstrates a fair behavior across all the diverse sensing modalities considered in the evaluation.

The *RMSE* values reported in Table 3 vouch that the highest accuracies are obtained by our proposal and SI. As for the SI pipeline, the high accuracy is likely to be due to the verification stage that refines the final alignment through ICP. Instead, we ascribe the high accuracy of our proposal to the large quantity of uniformly distributed correspondences that survive the filtering stages of the pipeline and jointly partic-

	N° Registrations				RMSE (mr)				CPU time (sec)			
	LRF	SI	4PCS	BSL12	LRF	SI	4PCS	BSL12	LRF	SI	4PCS	BSL12
Bunny (31/45)	27	20	16	14	1.09	0.74	3.77	2.37	0.74	338.53	1177.54	0.17
Dragon (108/190)	98	65	70	36	1.50	0.80	3.40	3.04	0.39	235.69	780.49	0.11
Armadillo (141/253)	118	72	95	39	0.94	0.95	3.23	2.25	0.20	72.50	453.07	0.12
Angels (18/28)	15	16	5	19	2.09	2.72	20.26	3.64	5.11	1074.31	386.86	0.58
Shell (822/4753)	620	--	104	646	1.71	--	22.32	3.24	1.21	--	258.07	0.48
Venus (256/1830)	207	148	51	160	2.25	2.91	15.94	4.54	0.68	414.2	393.91	0.25
MarioStereo (61/153)	58	44	57	38	4.10	4.03	6.44	9.66	0.65	136.28	19.94	0.10
SquirrelStereo (68/153)	44	34	33		2.77	3.43	4.74		0.54	112.68	358.96	
FrogStereo (83/210)	70	37	63		3.19	3.12	5.73		2.11	252.65	221.57	
MarioKinect (39/78)	30	26	32	9	2.32	1.56	3.33	8.49	0.08	115.04	951.67	0.01
SquirrelKinect (62/105)	34	33	33	9	2.24	1.33	2.04	5.41	0.03	88.32	1312.83	0.05
FrogKinect (96/190)	72	47	54	29	2.14	1.63	3.48	7.92	0.23	159.06	392.15	0.10
DuckKinect (61/120)	43	39	39	15	2.28	1.98	3.99	9.02	0.21	110.29	25.49	0.10
Room (55/153)	40	24	25	32	5.42	7.04	7.16	13.21	7.02	70.02	118.41	0.34
Amphora (63/91)	39	22	15		0.91	0.96	4.52		0.97	157.68	77.92	
Children (152/325)	123	96	81		0.82	1.07	6.25		1.82	65.13	64.69	
Neptune (34/105)	23	16	9		1.15	0.84	13.07		2.63	201.77	61.74	
Fish (22/45)	16	12	9		1.40	2.38	8.75		1.58	61.53	2.29	
MasterCylinder (245/378)	142	90	87		0.99	0.78	3.31		0.15	168.79	110.46	
OilPump (941/1540)	719	--	300		1.49	--	5.38		1.35	--	78.28	
Blade (29/55)	21	10	14		2.01	1.70	7.29		1.64	34.02	18.48	
WoodChair (58/105)	27	13	20		2.63	2.57	5.62		0.75	316.88	44.74	
Buste (85/120)	69	43	49		1.90	2.81	9.04		0.18	314.37	46.79	
Glock (16/28)	13	6	11		2.19	5.14	6.84		1.51	127.74	13.43	

Table 3: *N° Registrations, RMSE and CPU time of the four methods on the 24 datasets considered in the evaluation. Darker colors denote, respectively, a larger number of view pairs correctly registered, more accurate alignments and faster computations. For each dataset, the number of view pairs that share at least 10% of their surface as well as the number of tested view pairs, N, are reported between brackets.*

ipate in the final estimation of the rigid motion. This helps keeping the *RMSE* low across the entire surface acquired by a partial view.

Concerning computational efficiency, all the experiments were conducted on a PC equipped with a 3.50 GHz Intel i7 CPU and 16 GB RAM. The results show that the fastest method is BSL12, whereas, usually, 4PCS and SI spend conspicuously higher times to align views. However, BSL12 is optimized to work on multi-core architectures whilst our proposal exploits a single core so far. Moreover, a registration pipeline based on range images is inherently faster than one working on points clouds, due to the former deploying the lattice provided by the image to find the neighbours of a feature, the latter requiring a slower kd-tree search. Thus, the efficiency of our pipeline, which is comparable to that of BLS12, is due to the purposely devised feature extraction and inexpensive feature matching approaches.

Once all the pairwise rigid motions are estimated, it is possible to align the views in a unique reference frame. Based on the pairwise registrations provided by our pipeline, we exploit the framework of [BNSN12] to get a global, though coarse, reconstruction by determining a spanning tree where the edges join the view pairs that maximize the overlap area. We apply this process to the *Venus* dataset and two *Kinect* datasets. The results are depicted in Fig. 10 and Fig. 11. Due to both the acquisition quality of the *Venus* dataset and the accuracy of our pipeline, the views come out finely aligned, even though no ICP-based fine registration is run downstream. Conversely, for *DuckKinect* and *FrogKinect*, we then also run the *Scanalyze* tool to get a global refinement of the registration and the *Poisson Recon-*

struction algorithm [KBH06] to obtain the final 3D models. Although *Kinect* acquisitions are noisy and inaccurate, the results are worthy.

Figure 10: *Registration of the Venus de Milo by alignment of 61 views and about 50 million points.*

Figure 11: *Reconstructions of DuckKinect and FrogKinect. Top: initial disarranged views. Center: coarse reconstructions provided by our pipeline. Bottom: final meshes attained by refining coarse reconstructions by Scanalyze and then running Poisson Reconstruction.*

5. Final Remarks

Our evaluation neatly shows that, whereas 4PCS gets better results with lower-resolution data, BSL12 is suited for high-precision datasets and SI provides fairly stable performance, our approach attains considerable registration rates on any kind of dataset, regardless of the type of sensor used for acquisition. Furthermore, it runs in times comparable to those of BSL12, which exploits parallelism and works on range images. As for its limits, even though our proposal proves to successfully handle noisy data, it is not robust to the presence of outliers in the input data. In the *4-Points Congruent Set* paper, instead, the authors show that 4PCS easily deals with a broad percentage of outliers. Such weakness in our pipeline is due to the definition of the local reference frame that does not account for the presence of spurious points in the support. Another issue is the large number of parameters of our pipeline. However, only a bunch of them actually affects performance significantly. Indeed, the proposed evaluation suggests that most parameters may be left at their default values (Table 1), and only the support radius, R_x, adjusted by trial-and-error to optimize performance on unseen data.

Nonetheless, in order to both make its usage even easier and to improve performance, the pipeline may be equipped with an initial stage aimed at estimating some parameters automatically. For example, it may be possible to try to quickly estimate on-line the value of the support radius, R_x, based on a set of random probes, e.g. so as to optimize the trade-off between the information content associated with the basic shape cue deployed to match features, D, and computational efficiency.

In the reconstruction stage, to build the spanning tree including the best pairwise alignments, it is necessary to check for all the combinations of view pairs so as to find those showing high overlaps. This process can be costly, especially in case of datasets, like *Venus*, *Shell* and *OilPump*, comprising a large number of views. Even though the *Hough voting* stage is itself rather inexpensive, such a large number of runs may slow down reconstruction time notably. Therefore, it would be beneficial to be able to abort the registration before the Hough stage in case a view pair is unlikely to belong to the spanning tree. This may be done efficiently on the basis of the distribution of the scores resulting from the fast feature matching stage. More precisely, as a small $\widetilde{D}_{ij}$ is more likely to come from a good correspondence while a larger one will come from a wrong correspondence (see Fig. 4), we may analyze the distribution of $\widetilde{D}_{ij}$ in the current view pair, $p\left(\widetilde{D}_{ij}\right)$, so as to guess, e.g. based on the probability of $\widetilde{D}_{ij}$ to be small enough ($p\left(\widetilde{D}_{ij} < T\right)$), whether the pair provides enough good correspondences and, as such, is likely to belong to the spanning tree.

As already mentioned, the main bottleneck of the pipeline resides in the search for neighboring points, especially in the extraction of the spherical support used for the computation of the local reference frame. Indeed, a standard kd-tree extracts all the points inside the sphere, which mandates a further filtering operation to then select only those in the shell used to determine the tangential axis. To speed-up the

search, a dedicated indexing scheme may be devised to allow for a radius search that directly extracts only the useful points in the shell of the sphere.

References

[AMCO08] AIGER D., MITRA N. J., COHEN-OR D.: 4-points congruent sets for robust surface registration. *ACM Transactions on Graphics 27*, 3 (2008). 3, 8

[ART10] ALBARELLI A., RODOLÀ E., TORSELLO A.: Loosely Distinctive Features for Robust Surface Alignment. In *European Conference On Computer Vision* (2010). 2

[BM92] BESL P. J., MCKAY N. D.: A method for registration of 3-D shapes. *Transactions on Pattern Analysis and Machine Intelligence* (1992). 1

[BNSN12] BARIYA P., NOVATNACK J., SCHWARTZ G., NISHINO K.: 3D Geometric Scale Variability in Range Images: Features and Descriptors. *International Journal of Computer Vision 99*, 2 (2012), 232–255. 2, 12

[BSL11] BONARRIGO F., SIGNORONI A., LEONARDI R.: A robust pipeline for rapid feature-based pre-alignment of dense range scans. *International Conference on Computer Vision* (2011), 2260–2267. 2, 3, 8, 9

[BSL12] BONARRIGO F., SIGNORONI A., LEONARDI R.: Multi-view alignment with database of features for an improved usage of high-end 3D scanners. *EURASIP Journal on Advances in Signal Processing*, 1 (2012). 3, 8, 9

[CCFM08] CASTELLANI U., CRISTANI M., FANTONI S., MURINO V.: Sparse points matching by combining 3D mesh saliency with statistical descriptors. *Computer Graphics Forum* (2008). 2

[CJ97] CHUA C. S., JARVIS R.: Point signatures: A new representation for 3d object recognition. *International Journal of Computer Vision 25*, 1 (1997), 63–85. 2

[CL96] CURLESS B., LEVOY M.: A volumetric method for building complex models from range images. In *SIGGRAPH* (1996), pp. 303–312. 6, 9

[CYL98] CHUNG D., YUN I., LEE S.: Registration of multiple-range views using the reverse-calibration technique. *Pattern Recognition 31*, 4 (1998), 457–464. 1

[DNRR05] DAVIS J., NEHAB D., RAMAMOORTHI R., RUSINKIEWICZ S.: Spacetime stereo: A unifying framework for depth from triangulation. *Transactions on Pattern Analysis and Machine Intelligence. 27*, 2 (2005), 296–302. 9

[dSFMMH11] DOS SANTOS T. R., FRANZ A., MEINZER H.-P., MAIER-HEIN L.: Robust multi-modal surface matching for intra-operative registration. *25th International Symposium on Computer-Based Medical Systems 0* (2011), 1–6. 2

[FA96] FELDMAR J., AYACHE N.: Rigid, affine and locally affine registration of free-form surfaces. *International Journal of Computer Vision* (1996). 2

[FB81] FISCHLER M., BOLLES R.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* (1981). 2

[FHK*04] FROME A., HUBER D., KOLLURI R., BÜLOW T., MALIK J.: Recognizing objects in range data using regional point descriptors. In *European Conference On Computer Vision* (2004), vol. 3, pp. 224–237. 2

[Hor84] HORN B. K. P.: Extended gaussian images. *Proceedings of the IEEE*, 12 (1984). 1

[Hor87] HORN B. K. P.: Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A 4*, 4 (1987), 629–642. 5

[JH97] JOHNSON A. E., HEBERT M.: Surface registration by matching oriented points. *International Conference on 3D Digital Imaging and Modeling* (1997). 3, 8

[JH99] JOHNSON A. E., HEBERT M.: Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence 21* (1999), 433–449. 2

[KBH06] KAZHDAN M., BOLITHO M., HOPPE H.: Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing* (2006), Eurographics Association, pp. 61–70. 13

[KK12] KULKARNI N., KUMAR S.: Vote based correspondence for 3D point-set registration. *Indian Conference on Computer Vision, Graphics and Image Processing* (2012). 2

[KPW*10] KNOPP J., PRASAD M., WILLEMS G., TIMOFTE R., GOOL L. V.: Hough transform and 3d surf for robust three dimensional classification. In *European Conference On Computer Vision* (2010), pp. 589–602. 2

[LG05] LI X., GUSKOV I.: Multi-scale Features for Approximate Alignment of Point-based Surfaces. *Eurographics Symposium on Geometry Processing* (2005). 2

[LW09] LIYING W., WEIDONG S.: A review of range image registration methods with accuracy evaluation. *Urban Remote Sensing Joint Event* (2009), 1–8. 1

[Mas09] MASUDA T.: Log-polar height maps for multiple range image registration. *Computer Vision and Image Understanding 113*, 11 (2009), 1158–1169. 2

[MBO06] MIAN A., BENNAMOUN M., OWENS R.: A novel representation and feature matching algorithm for automatic pairwise registration of range images. *International Journal of Computer Vision 66*, 1 (2006), 19–40. 2

[MPD06] MAKADIA A., PATTERSON A., DANIILIDIS K.: Fully Automatic Registration of 3D Point Clouds. In *Conference on Computer Vision and Pattern Recognition* (2006). 1

[MW08] MÁRQUEZ M. R. G., WU S. T.: An automatic crude registration of two partially overlapping range images. *Brazilian Symposium on Computer Graphics and Image Processing* (2008), 245–252. 2

[PD11] PETRELLI A., DI STEFANO L.: On the repeatability of the local reference frame for partial shape matching. *International Conference on Computer Vision* (2011), 2244–2251. 2

[PDS12] PETRELLI A., DI STEFANO L.: A repeatable and efficient canonical reference for surface matching. In *Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission* (2012), pp. 403–410. 2, 3, 4, 6

[RBB09] RUSU R. B., BLODOW N., BEETZ M.: Fast Point Feature Histograms (FPFH) for 3D registration. *International Conference on Robotics and Automation* (2009), 3212–3217. 2

[RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the ICP algorithm. *International Conference on 3D Digital Imaging and Modeling* (2001). 1

[SA01] SUN Y., ABIDI M. A.: Surface matching by 3d point's fingerprint. *International Conference on Computer Vision 2* (2001), 263–269. 2

[SEE*12] STURM J., ENGELHARD N., ENDRES F., BURGARD W., CREMERS D.: A benchmark for the evaluation of rgb-d slam systems. In *International Conference on Intelligent Robot Systems* (2012). 9

[SHT09] SEGAL A., HAEHNEL D., THRUN S.: Generalized-ICP. In *Robotics: Science and Systems* (2009). 10

[SM92] STEIN F., MEDIONI G.: Structural indexing: Efficient 3-d object recognition. *Transactions on Pattern Analysis and Machine Intelligence 14*, 2 (1992), 125–145. 2

[SMFF07] SALVI J., MATABOSCH C., FOFI D., FOREST J.: A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing 25* (2007), 578–596. 2

[STD13] SALTI S., TOMBARI F., DI STEFANO L.: A performance evaluation of 3d keypoint detectors. *International Journal of Computer Vision 102* (2013), 198–220. 5, 7, 8

[TCC98] TAREL J., CIVI H., COOPER D.: Pose estimation of free-form 3D objects without point matching using algebraic surface models. *Workshop Model Based 3D Image Analysis* (1998). 1

[TCL*13] TAM G. K. L., CHENG Z.-Q., LAI Y.-K., LANGBEIN F. C., LIU Y., MARSHALL D., MARTIN R. R., SUN X.-F., ROSIN P. L.: Registration of 3D point clouds and meshes: a survey from rigid to nonrigid. *Transactions on Visualization and Computer Graphics 19*, 7 (2013), 1199–217. 2

[TDS10] TOMBARI F., DI STEFANO L.: Object recognition in 3d scenes with occlusions and clutter by hough voting. In *Proceedings of the 2010 Fourth Pacific-Rim Symposium on Image and Video Technology* (2010), pp. 349–355. 2, 5

[TSD10a] TOMBARI F., SALTI S., DI STEFANO L.: Unique shape context for 3d data description. In *ACM Workshop on 3D Object Retrieval* (2010), pp. 57–62. 2

[TSD10b] TOMBARI F., SALTI S., DI STEFANO L.: Unique signatures of histograms for local surface description. In *European Conference On Computer Vision* (2010), pp. 356–369. 2

[YM92] Y. C., MEDIONI G.: Object modelling by registration of multiple range images. *Image and Vision Computing* (1992), 2724–2729. 1

[ZBH12] ZAHARESCU A., BOYER E., HORAUD R.: Keypoints and Local Descriptors of Scalar Functions on 2D Manifolds. *International Journal of Computer Vision* (2012). 2, 7

[ZCS03] ZHANG L., CURLESS B., SEITZ S. M.: Spacetime stereo: Shape recovery for dynamic scenes. In *Conference on Computer Vision and Pattern Recognition* (2003). 9

[Zho09] ZHONG Y.: Intrinsic shape signatures: A shape descriptor for 3D object recognition. In *International Conference on Computer Vision Workshops: 3DRR* (2009). 2, 7