Relative Scale Estimation and 3D Registration of Multi-Modal Geometry Using Growing Least Squares

Nicolas Mellado, Matteo Dellepiane, and Roberto Scopigno

Abstract—The advent of low cost scanning devices and the improvement of multi-view stereo techniques have made the acquisition of 3D geometry ubiquitous. Data gathered from different devices, however, result in large variations in detail, scale, and coverage. Registration of such data is essential before visualizing, comparing and archiving them. However, state-of-the-art methods for geometry registration cannot be directly applied due to intrinsic differences between the models, e.g., sampling, scale, noise. In this paper we present a method for the automatic registration of multi-modal geometric data, i.e., acquired by devices with different properties (e.g., resolution, noise, data scaling). The method uses a descriptor based on Growing Least Squares, and is robust to noise, variation in sampling density, details, and enables scale-invariant matching. It allows not only the measurement of the similarity between the geometry surrounding two points, but also the estimation of their relative scale. As it is computed locally, it can be used to analyze large point clouds composed of millions of points. We implemented our approach in two registration procedures (assisted and automatic) and applied them successfully on a number of synthetic and real cases. We show that using our method, multi-modal models can be automatically registered, regardless of their differences in noise, detail, scale, and unknown relative coverage.

Index Terms—Multi-modal data, 3D registration, multi-scale descriptors

1 Introduction

In this paper, we focus on the registration of 3D multi-modal data describing surfaces, i.e., on point clouds or meshes generated by different acquisition devices (e.g., laser scans, depth cams, multi-view stereo reconstruction) or modeling tools. Indeed, acquisition devices are today ubiquitous and affordable, making the surface acquisition more and more popular. This is a consequence of the evolution of manual and assisted image modeling tools, of the development of automatic and robust methods for multi-view stereo reconstruction, and of the availability of low cost depth cameras like Kinect. As a consequence, multi-modal registration is today a common issue when working with 3D objects: a typical scenario is the registration of a low-resolution point-cloud to a high resolution mesh generated from 3D scanning.

The main difficulty when working with multi-modal data is to be robust to a large heterogeneity of geometric properties, e.g., noise, sampling, and scaling. Many approaches have been already proposed to register 3D surfaces, however they are mainly devoted to range maps alignment or non rigid registration, and usually assume almost-uniform geometric properties within the data. In practice they do not prove to be effective for multi-modal data, due to differences in terms of:

- *Density and detail amount*: surface details may not be present on some types of data, and sampling density can vary a lot.
- *Scale*: data coming from manual or assisted modeling, or structure from motion are in an arbitrary scale. Hence, a scale factor has to be estimated in addition to roto-translation.
- *Noise and deformation*: non-uniform noise and deformations are common in data which do not come from 3D scanning.
- *Connectivity*: multi-view stereo methods may not be able to provide a surface, but only a point cloud, hence the connectivity information may not be available.
- *Overlap*: the surface covered by the data is usually not known in advance, so no information about the amount of overlap is available.

A similar problem is faced by the Medical Imaging community, which have to analyze, register and process data acquired by different devices [2], [3], [4]. The main motivation is here to acquire different properties of the matter composing a subject for a given application purpose, and combine them in a meaningful manner. To do so, methods are designed to process low resolution 3D images, where the information is stored by voxel: algorithms take benefits of the regular structure (3D grid) as a support for computations. In our case, the lack of a regular spatial structure prevents to apply such methods on detailed meshes, and even less on unstructured data like point-clouds.

In this paper, we present a method for the registration of multi-modal geometric data, based on the Growing Least Squares descriptor (GLS) [5] and illustrated in Fig. 1. This approach defines a meaningful scale-space representation for point-clouds, and provides both local and global descriptions

- *N. Mellado is with the Université de Toulouse, UPS, IRIT, Toulouse, France. E-mail: nmellado0@gmail.com.*
- *M. Dellepiane and R. Scopigno are with the Visual Computing Lab, ISTI-CNR, Pisa, Italy. E-mail: {matteo.dellepiane, R.Scopigno}@isti.cnr.it.*

Manuscript received 29 May 2015; revised 8 Oct. 2015; accepted 18 Nov. 2015. Date of publication 8 Dec. 2015; date of current version 3 Aug. 2016. Recommended for acceptance by A. Shamir.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2015.2505287

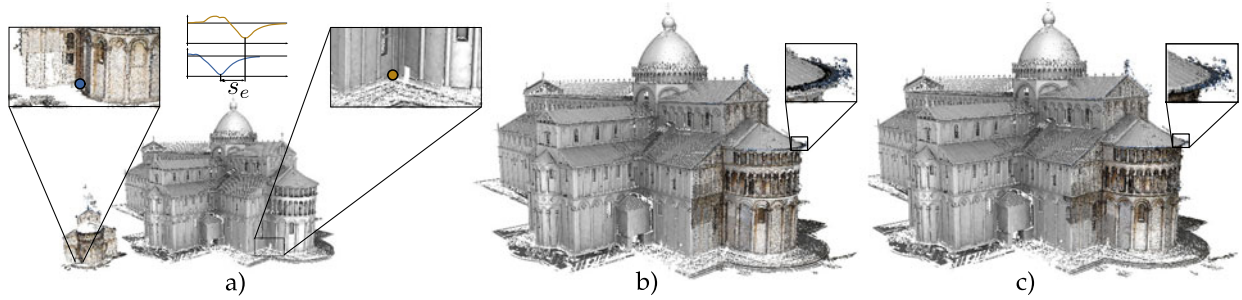


Fig. 1. We present a method for the automatic registration of 3D *multi-modal* discrete surfaces, ie. generated by different acquisition devices. a) Starting from a reference model (acquired by LIDAR) defining the scale of the scene, our approach automatically estimates a local relative scale s_e of a dense mesh (from multi-view stereo) by matching and comparing a descriptor based on Growing Least Square. b) Point-wise relative scales are used to register the two models using a RANSAC variant. c) Alignment and scale are refined using [1].

of the geometry. The need to work with a meaningful multi-scale description is illustrated in Fig. 2: a point $\mathbf{p}$ is within shapes of three different sizes: a local bump, the woman shoulder and the complete torso. The associated scales, respectively s_1, s_2, and s_3, can be detected using GLS and used for further processing, e.g., registration. The scale is here defined as the size of an Euclidean neighborhood around $\mathbf{p}$.

Based on this analysis framework, we present a new approach to estimate the relative scale factor between two multi-modal models. In opposite to previous work, our approach is robust to intra and inter-model variations of sampling and noise, and doesn't require an intermediate surface reconstruction step. As a consequence, our approach can be applied directly on large and raw point clouds, just after registration and without any pre-process. Our contributions can be summarized as follow:

- We present a new robust point-wise *scale estimation* and *multi-scale comparison operator*, which is invariant to relative scale factors, fast to evaluate and robust to typical acquisition artifacts.
- We demonstrate the efficiency and robustness of this operator through two practical registration frameworks (assisted or automatic), easy to implement and successfully applied on multi-modal models composed of millions of points at arbitrary scales (see example in Fig. 1).

2 Related Work

Three-dimensional geometry registration transforms multiple 3D datasets in a common reference system. The alignment can be obtained using a rigid or non-rigid transformation. Please refer to [6], [7], [8] for surveys covering the main issues and methods related to this task.

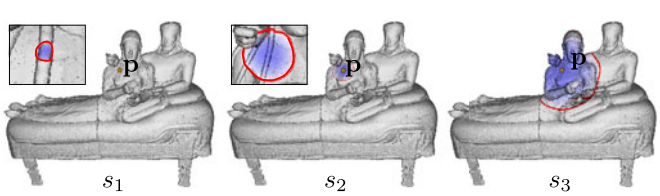


Fig. 2. A point $\mathbf{q}$ on the model SPOUSE$_{\text{SCAN}}$, which belongs to structures defined at three different scales (s_1, s_2, s_3, shown as red circles): a local bump, the shoulder and the entire torso. Using a multi-scale signature to represent this point and its neighborhood is necessary to get a meaningful description of both local and global shape properties.

In this paper, we focus on the registration of data describing rigid objects. The alignment of geometry data from rigid objects is usually split in two steps: a *rough* alignment, that estimates an initial registration, and a *fine* alignment, which refines the registration starting from the result of the previous step. The *fine* alignment step can be performed in a fully automatic way, and the proposed solutions, e.g., instantaneous kinematic [9] and Iterative Closest Point (ICP) with its variants [1], [10], [11], [12], proved to be robust and reliable. In the opposite, automatic *rough* alignment is a more challenging task, especially if no assumptions on the input data are done. From a practical point of view, current software for 3D scanning achieves the alignment of range maps using markers or user intervention.

The goal of our approach is to automatically find a *rough* alignment for multi-modal data, with the possibility to *refine* it in a further step. For this reason, this section will focus on a broad overview of the existing alignement approaches, and stressing their applicability to multi-modal data.

2.1 Point Clouds Registration

The problem of registering acquired point clouds has been thoroughly studied in the last few years. Local shape descriptors (see Heider et al. [13] for a survey) can be used to match points with similar signatures, and obtain an initial registration, which can be refined using ICP. Some examples of local descriptors include Spin Images [14], feature lines [15], feature histograms [16], and methods based on SIFT and their variants [17], [18]. All of these methods are limited by their locality, so that data with different density or level of detail cannot be treated properly. An interesting alternative would be to use Integral Invariants as multiscale signatures and use them for matching [19]. However such approaches require to build a volumetric representation of the data, which would be challenging because of the noise and holes potentially present in our data.

Other approaches are based on the correlation of Extended Gaussian Images (EGI) in the Fourier domain [20], or perform the optimization directly on the affine space [21]. Krishnan et al. [22] and Bonarrigo and Signoroni [23] proposed a framework to perform the optimization on the manifold of rotations through an iterative scheme based on Gauss-Newton optimization method. None of the above methods is very robust against noise or outliers.

A very interesting alternative is the 4-Point Congruent Set (4PCS) [24] which explores the transformation space

between two clouds by matching almost-planar four-point quadrilaterals. The proposed scheme can run without shape descriptor, is robust to noise and outliers, and runs with a lower complexity than RANSAC, respectively performing in quadratic versus cubic time according to the number of points. 4PCS can theoretically be extended to estimate a relative scale factor, but in that case the size of the explored congruent set makes the procedure impracticable. The recent Super4PCS variant [25], tailored for rigid transformation estimation, uses efficient indexing techniques to run in linear time in number of point.

2.2 Multi-Modal Geometry Registration

The registration of multi-modal geometric data is usually performed in a semi-automatic fashion. A user provides an initial alignment by manually solving the problem of the initial roto-translation and scaling of the data, or by specifying point-wise correspondences between objects. The analysis of data coming from different devices is applied in methods which aim at automatically align groups of images in a geometry. For instance in the work by Pintus et al. [29], a point cloud generated by structure from motion is used to align a set of images to a different point cloud. Nevertheless, even in this case the initial alignment is estimated using user input. Alternatively, if images were used for geometry generation, they can be analysed to find common features and calculate the 3D registration [30].

The factors described in Section 1 (e.g., scale difference, data density and noise *in primis*) prevent from using most of the aforementioned approaches. An evaluation by Kim and Hilton [31] shows that statistical local descriptors perform poorly in the registration of multi-modal data.

An increasing amount of techniques have been proposed recently to address this issue. Lee et al. [32] proposed a method to align 3D scans by analyzing and matching surfaces in the 2D parametric domain. While able to cope with scale differences, the solution is not tested on multi-modal data, and requires parametrized surfaces. Quan and Tang [33] proposed a scale invariant local descriptor applied for 3D part-in-whole matching for mechanical pieces. Unfortunately, the method may be applied only on meshes, and the robustness to noise and to lack of features (i.e., the strong edges of mechanical pieces) is unclear. Rodola' et al. [26] use a game-theoretic framework with scale-independent descriptors in the context of object recognition. While independent of scaling, the method relies on similar geometries, and it can be applied only on continuous surfaces or extremely dense point clouds. A similar approach, that combines local descriptors at different scale to extract and compare the so-called *keyscale* of a model has been recently proposed by Lin et al. [28]. The goal of this method is to estimate the relative scale only, so that ICP can be applied to find the alignment. The proposed descriptor proved to be robust to noise. Nevertheless, it seems to be effective only if a major overlap between the models is present.

In the context of an automatic image registration system, Corsini et al. [27] proposed an automatic method for the alignment of a point-cloud acquired by Structure-from-Motion on another 3D model. The method is based on an extension of 4PCS, and it makes use of the Variational Shape Approximation algorithm [34] to reduce the complexity of the 4PCS when scale has to be estimated and no previous information about overlapping is known. Moreover, this approach requires to reconstruct or at least approximate the input object surface, which can be tedious in some cases. In the context of feature independent methods, a parameters-free framework to fit models to contaminated data was applied also on an example of scale-independent 3D similarity transformation [35].

2.3 Multi-Scale Geometry Analysis

The analysis of 3D objects at multiple scales has been widely studied recently. It has been mainly focused on shape retrieval in 3D databases [36] and matching of deformable objects [37]. Most of the techniques recently presented focus on the intrinsic geometry properties such as the eigenfunctions of the Laplace-Beltrami operator [38], [39], and the Heat Kernel Signatures [40], [41]. Originally limited to meshes, recent approaches have been proposed to extend diffusion and geodesic distances to point clouds [42], [43]. These approaches do not fit our requirement since they cannot automatically detect the scale associated to detected features [44] and thus do not help to estimate the scale between two models. Moreover, they require to solve the diffusion equation globally on the entire object, which makes their use impractical with acquired 3D objects defined by millions of points. An alternative is to evaluate the diffusion on a subsampled geometry to speed up computation. However, in this case the details removed from the original model cannot be caught by the multi-scale signatures.

The goal of *scale-space* techniques is to analyze a signal at different scales to discover its geometric structure [45]. A well known use of this theory is the feature detection stage of the Scale-Invariant Feature Transform (SIFT) [46]. These approaches rely on the existence of a parametrization, which is used to compute the spatial derivatives of a signal and extract its relevant structures at multiple scales. Methods have been proposed to adapt SIFT-point detection to 3D by extending image-based techniques either to voxel grids [47], [48], or locally on surfaces using mesh connectivity [49], [50], [51]. In our case we need to work with point-clouds, which makes such techniques unusable without prior remeshing. We refer the reader to the recent work of [44] for a practical and up-to-date comparison of mesh-based *scale-space* techniques. Recently, Mellado et al. have proposed a technique called *Growing Least Squares* [5], which aims at extending the *scale-space* formalism to point-set surfaces using implicit kernels evaluated at growing scales. This approach is computed locally at any location on the object, does not require any parametrization, supports arbitrary scale sampling and can be used to represent and match features using a robust multi-scale geometry descriptor.

2.4 Discussion

The registration of multi-modal data shares most of the main steps with the usual registration of 3D data. During the *rough* registration step, a—potentially large—geometric transformation is computed in order to transform one model and match the other. In case the scale is unknown, as when considering multi-modal data, one need to estimate both the rigid transformation and the scale factor to match

TABLE 1
Comparison of Existing Registration Techniques Regarding Their Robustness to Multi-Modal Data Specificities

Differences in ...	BBox fitting	Game-Theoretic framework [26]	SIFT [17], [18]	Corsini et al. [27]	Keyscale [28]	Our method
Density and detail amount	V	V?	-	V	V	V
Noise and deformation	V	-	V?	V	V	V
Scalability	V	V	V	V	-	V
Point-clouds	V	-	-	-	V	V
Low overlap	-	V	V	V	V?	V

In addition to the variations of properties that may occur between the models (e.g., in term of sampling or noise), registration techniques need to handle potentially large point-clouds with unknown overlap.

the models. For very specific case, e.g., partial shape matching with very important overlap, one can estimate the transformation by fitting the two models bounding boxes. In other cases, it becomes necessary to explore the transformation space, preferably by finding correspondences between the models. In real world scenarios, finding these correspondences is a critical step, unlocking at the same time the estimation of the relative scale factor and the rigid transformation. For these reasons we designed our method as a point-wise comparison operator, allowing to estimate a relative scale factor between two points. We argue that when a method is able to find the scale between two models, and can also be used to compute the rigid transformation to align them.

State-of-the-art registration methods may require *ad-hoc* modifications to estimate a relative scale factor and handling multi-modal data specificities, as described in Section 1. We compare in Table 1 the robustness of techniques estimating both scaling and alignment regarding these properties. We added a row on methods *scalability*, in order to emphasis that modern acquisition methods can generate massive amounts of data, which should be taken into account during registration. The "V?" symbol indicates that the method may have the possibility to deal with an issue, but hasn't been tested. From our experiments and original papers conclusions, none of the state-of-the-art method is able to cope with all the issues related to multi-modal data. Our method on the other hand, is able register models from real-world multi-modal datasets, as shown in Section 5.

3 Overview

In this paper we propose a new practical registration technique for 3D multi-modal models, robust to the characteristics described in Section 1. It is designed as follows:

- We use the GLS descriptor to characterize *point-based* data, and use logarithmic scale-space to get scale-invariant signature. This description is robust to *noise*, a realistic amount of *outliers*, and provides a multi-scale point-wise comparison operator robust to *variation of density* and *detail amount*.
- We propose a new approach to compare *point-wise* multi-scale profiles and estimate the *relative scale factor* between points (see Fig. 1a). The resulting operator is easy to implement and fast to evaluate. The amount of relative scaling that can be handled by our approach depends only on the scale-space sampling domain.
- We demonstrate the robustness of our scale estimation to match and estimate scales between points of two input models, and guide global matching techniques (e.g., manually assisted and fully automatic RANSAC) to estimate a rigid transformation between them (see RANSAC results in Fig. 1b).

The paper is structured as follow: we first focus on the *point-wise* similarity and scale estimation in Section 4, and then present how we use it to register multimodal data in Section 5. Both sections present related contributions and associated results and evaluations.

4 Point-Wise Relative Scale Estimation

This section is organized as follow: we first recall basics concepts related to the Growing Least Squares descriptor, then we present how we extend the GLS comparison operator to estimate a relative scale factor between two points. Then, we evaluate the robustness of our point-wise similarity and scale estimation under variable amount of noise, details, outliers and variations of scales and density.

4.1 Background

The key idea of the GLS approach is to perform a scale-space analysis of point-set surfaces by means of continuous algebraic fits. More specifically, a scale-space is built through least-square fits of an algebraic sphere onto neighborhoods of continuously increasing sizes. The use of an algebraic surface ensures robust fits even at large scales [52] and yields a rich geometric descriptor with only a few parameters, called the *GLS descriptor* [5]. The continuity of the fitting process through scales provides for a stable and elegant analysis of geometric variations, called the *GLS analysis*.

The GLS descriptor is composed of three geometric parameters (illustrated in Fig. 4) describing the geometry surrounding an arbitrary location $\mathbf{p}$: the algebraic distance τ_s between $\mathbf{p}$ and the fitted primitive, the unit normal vector $\boldsymbol{\eta}$, and the mean curvature κ_s, where s is the evaluation scale defining the size of the neighborhood used to compute these values. We combine these three parameters with the scale invariant fitness value $\varphi \in [0\ ;\ 1]$, computed as the fitting residuals [52]. Parameters τ_s and κ_s depend on the size of the object and are not reliable to compare multi-modal data. According to [5], one may use their scale-invariant counterparts $\tau = \frac{\tau_s}{s}$, and $\kappa = \kappa_s * s$. Examples of normalized profiles are shown in Fig. 3b). In the following we note $GLS(\mathbf{p}, s) = \left[\tau\ \boldsymbol{\eta}^T\ \kappa\ \varphi\right]^T$ to refer to the scale invariant Growing Least Squares descriptor of point $\mathbf{p}$ at scale s. We refer

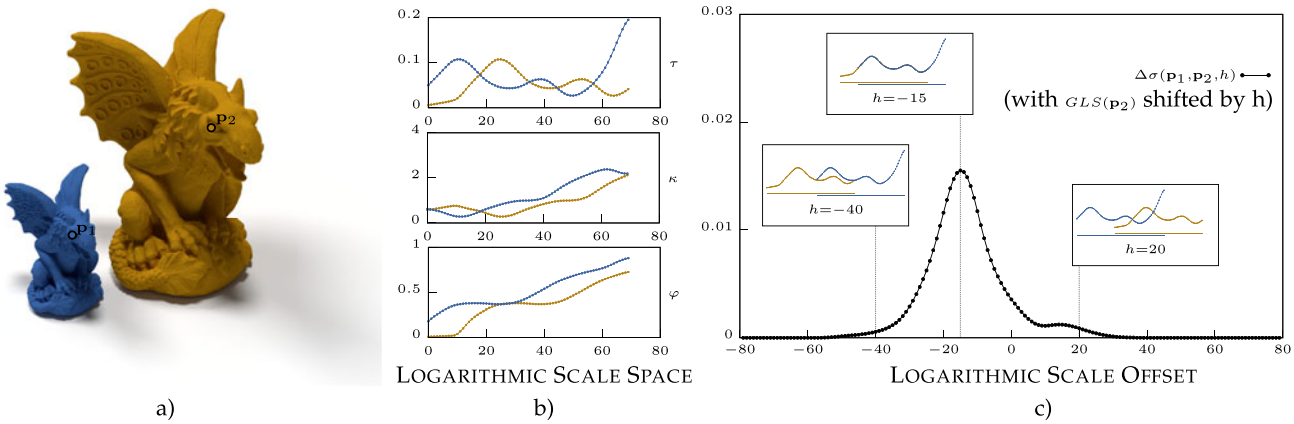


Fig. 3. Overview of our scale estimation. (a) Models GARGSCALED (blue) and GARGOYLE (gold). Their relative scale of 2 leads to a translation of the scale-invariant GLS profiles (τ, κ, φ) in logarithmic scale-space (log base $= 1.05$), here shown for $\mathbf{p}_1$ and $\mathbf{p}_2$ (b) (colors are consistent). We convolve the two profiles, measure their similarity Δ (see Eq. (2)), and extract the relative offset h giving the best score (c), here -15, giving an estimated scale $\frac{1}{1.05^{-15}} = 2.08$ (see Eq. (3)).

to the original paper for a more in depth presentation and evaluation of the GLS framework.

According to the original paper, two descriptors computed at different locations $\mathbf{p}, \mathbf{p}'$ and scales s, s' can be compared using the dissimilarity function δ, defined as

$$\delta(\mathbf{p}, s, \mathbf{p}', s') = w_\tau(\tau(s) - \tau'(s'))^2 + w_\kappa(\kappa(s) - \kappa'(s'))^2 + w_\varphi(\varphi(s) - \varphi'(s'))^2, \quad (1)$$

where $\delta = 0$ means a perfect match. GLS descriptors are by definition translation and scale invariants. The rotation invariance is achieved by ignoring the unit normals $\boldsymbol{\eta}$ and $\boldsymbol{\eta}'$. In this paper we used $w_\tau = w_\kappa = w_\varphi = 1$.

This measure can be integrated over a scale domain $[a, b]$ to compare multi-scale profiles, leading to the dissimilarity function

$$\Delta(\mathbf{p}, \mathbf{p}') = \frac{1}{b-a}\int_a^b \delta(\mathbf{p}, s, \mathbf{p}', s)\, ds. \quad (2)$$

Note that the integral is normalized in order to be invariant to the domain size.

4.2 Point-Wise Relative Scale Estimation and Similarity

In the original GLS paper, authors propose to compare discrete multi-scale signatures by summing their dissimilarity Δ over a discrete scale intervals. This approach is valid only when the scale sampling is consistent between the two descriptors, and thus cannot be used directly in our case. As defined in Section 1, multi-modal data can be obtained at arbitrary scales, making point-wise multi-scale signature comparisons challenging. Anyway, we propose in this paper to use the GLS to estimate a relative scale factor between two models.

4.2.1 Scale Estimation

According to Bronstein and Kokkinos [41], a scale-space can be constructed to allow scale-invariant multi-scale signature comparison, eg. Scale-Invariant Heat Kernel Signatures (SI-HKS). The key idea is to build a *"logarithmically sampled scale-space in which shape scaling corresponds, up to a multiplicative constant, to a translation"*. The goal of SI-HKS is to provide a scale-invariant signature comparison, hence the translation between the two profiles is undone using Fourier analysis and the corrected profiles compared.

One option could be to use the translation between SI-HKS profiles to estimate a relative scale factor between two points. However, in this approach the notion of scale is defined as a diffusion time, and as far as we know, there is no way to convert a difference of diffusion time to an actual distance in Euclidean space.

We propose to solve this issue by computing GLS descriptors in logarithmic scale-space instead of SI-HKS signatures, and use profile translations to estimate a relative scale factor between two points. Thanks to its scale-invariant formulation and as shown in Fig. 3, GLS profiles exhibit the same behavior than SI-HKS signatures when computed on similar objects with different scales: profiles are shifted along the scale dimension. Assuming that the two profiles are computed using a logarithmic scale sampling with basis m, the relative scale s_e between the two models can be retrieved as

$$s_e = \frac{1}{m^h}, \quad (3)$$

where h the translation between two profiles in logarithmic space. An example of shifted descriptors is shown in Fig. 3b).

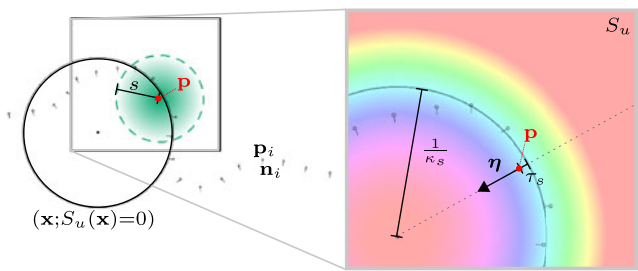


Fig. 4. Geometric meaning of the three geometric components of the GLS descriptor, illustrated in the 2D case: τ_s is the algebraic distance between $\mathbf{p}$ and the fitted primitive, $\boldsymbol{\eta}$ is the unit normal vector, and κ_s the mean curvature, computed at scale s. Figure courtesy Mellado et al. [5].

4.2.2 Scale-Invariant Similarity

The problem we now need to solve is how to match shifted profiles and compute the associated offset in logarithmic space. When profiles are shifted, one solution is to use the Fourier transform to undo the translation in scale-space and then compute profile similarity, like it is done with SI-HKS. However in our case, the GLS descriptor is a multi-dimensional vector, and our goal is to take into account each of its component to estimate the offset and compute the similarity.

Our approach is illustrated in Fig. 3, and allows us to estimate the scale offset between two descriptors and compute their dissimilarity at the same time. To do so, we propose to *convolve* the two sets of profiles. For each convolution step, we compute a similarity measure σ (see Eq. (4)), and select the best score configuration. The associated logarithmic shift h can then be used in Equation (3) to estimate the relative scale between the two points. We define the similarity measure σ returning 0 for incompatible descriptors and 1 for a perfect matches as

$$\sigma(\mathbf{p}, s, \mathbf{p}', s') = 1 - \tanh(\alpha * \delta(\mathbf{p}, s, \mathbf{p}', s')), \tag{4}$$

where tanh is the hyperbolic tangent. The hyperbolic tangent is used to map the values computed by δ from $[0, +\infty]$ to $[0, 1]$, in order to avoid large dissimilarity values and facilitate further processing [5]. The input range of the hyperbolic tangent function is adjusted using α, we used $\alpha = 4$ in all our experiments ($\sigma \approx 0$ when $\delta > 0.5$)

Given this normalized measure, we can now define our discrete scale-invariant similarity and shift estimation operator, as

$$\Delta_\sigma(\mathbf{p}, \mathbf{p}', h) = \frac{1}{2I} \sum_{i=-I}^{I} \sigma(\mathbf{p}, h - i, \mathbf{p}', h), \tag{5}$$

where $2I$ is the length of the overlapping interval between two descriptor for a logarithmic offset h between the two descriptors. Note that we normalize the sum result by the length of the associated interval. This normalization has two purposes: firstly it ensures to produce exactly the same results as the original dissimilarity Δ (see Eq. (2)) when $h = 0$. Secondly, it avoids to have a dependency to the interval $2I$, which would naturally favoring solutions with large overlap, i.e., when $h \to 0$. The relative scale between descriptors is then estimated using Equation (3), with h computed as

$$\arg\max_h \Delta_\sigma(\mathbf{p}, \mathbf{p}', h). \tag{6}$$

We illustrate in Fig. 3 how we compute the best offset by convolving the two descriptors in logarithmic scale-space.

It is important to remark that convolving two GLS descriptors and picking the higher similarity value does not guarantee to estimate the real scale factor between two points, as illustrated in Fig. 5a. However, we observed in practice that this kind of situation can usually be avoided, as shown in next section. In addition, it is usually safer to ignore values of Δ_σ generated from a very small overlap intervals to avoid instabilities and non-representative results.

More in general, four parameters are related to the GLS profiles: the logarithmic base, the number of scale samples to estimate, the minimum and the maximum scale to estimate. Setting three of them automatically sets the fourth. Setting the parameters may influence the performance of the descriptor (i.e., the minimum scale can be too big to account for small details), but their relation with the shape of the object is straightforward. A systematic evaluation of the robustness to parameters changes is presented in the next section. Moreover, we observed during our experiments that a set of parameters is enough to cope with very different cases (see Section 5.2), and can also be adjusted manually in an intuitive way (see Section 5.1).

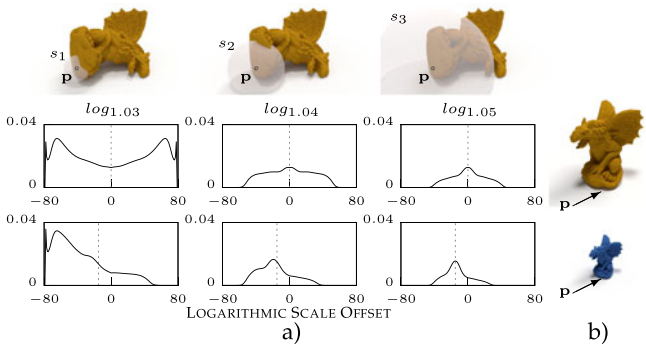


Fig. 5. Point-wise relative scale estimation between scaled versions of the same cloud, for the point $\mathbf{p}$. Top row is without scaling, bottom row with a 0.5 scaling factor. Three different scale samplings are studied, starting from the same minimum scale, but with different log bases, leading to three different scale ranges $[1, s_1]$, $[1, s_2]$, and $[1, s_3]$. The expected positions of the Δ_σ maximum is shown as vertical dashed lines.

4.3 Evaluation

In this section we evaluate the robustness of the point-wise relative scale estimation technique described in the previous section, against different geometric configurations and artifacts that can be found in multi-modal data. We evaluate our approach by varying the scale-space sampling and measuring its impact on the estimation. We also highlight geometric configurations that could break the scale estimation, and show how they can be detected and thus avoided. We used generic models in this section to make further comparison easier, and we refer the reader to Section 5 for more results on acquired objects.

4.3.1 Evaluation Point and Scale Range

The efficiency of the scale estimation is directly related to the properties of the compared GLS descriptors. The fact that a descriptor could be properly used in Eq. (6) is strongly influenced by the quantity of *significant* geometric configurations surrounding the evaluation point within the studied scale range.

Fig. 5 illustrates these conditions. The point $\mathbf{p}$ stands in a part of the object which is locally planar. Hence, computing Eq. (6) fails essentially when the scale range used for computation is too small. When the scale range increases, additional geometric details are added, improving the scale estimation. This is because local, regional and global configurations are combined during the estimation. This also shows that the position of the evaluation points is important. When a manual choice is needed, points exhibiting geometric details at different scales could be preferable. If a

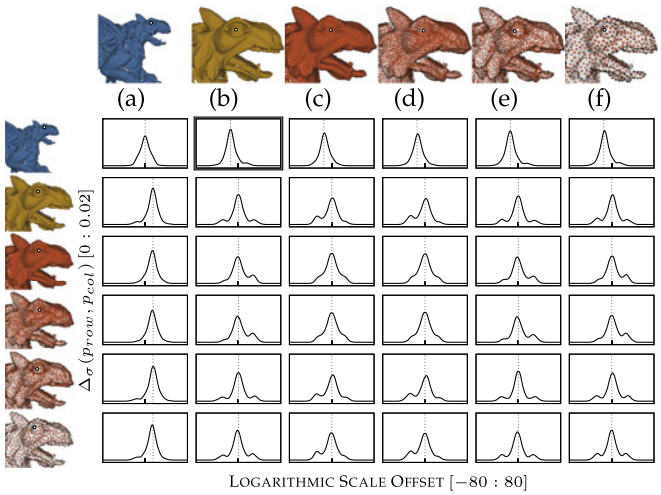


Fig. 6. GLS convolution between models $\text{GARG}_{\text{SCAN}}$ (a), GARGOYLE (b), $\text{GARG}_{\text{B500k}}$ (c), $\text{GARG}_{\text{B120k}}$ (d), $\text{GARG}_{\text{P66k}}$ (e) and $\text{GARG}_{\text{P10k}}$ (f). The selected point is shown as white dot, and selected manually in the eye of the gargoyle. For the sake of clarity, we show the three last point-clouds on top of a reference mesh, but only the visible points are used for computation. Expected maxima position are shown as vertical dashed lines, see Table 1 in additional materials for numerical values, maximum error reported: 2 units. See zoom of $\Delta_\sigma(a, b)$ in Fig. 3c.

candidate point has to be chosen automatically, a proper analysis of its GLS profiles could provide feedback about its quality. Additional comments on this can be found in Section 5.

4.3.2 Noise and Spatial Sampling

We evaluated the stability of our scale estimation by comparing models with almost the same geometry but with different samplings: GARGOYLE and $\text{GARG}_{\text{SCALED}}$ (see Fig. 3), $\text{GARG}_{\text{B500k}}$ and $\text{GARG}_{\text{B120k}}$ (from Berger et al. [53][1]), $\text{GARG}_{\text{P66k}}$ and $\text{GARG}_{\text{P10k}}$ (subsampled versions of GARGOYLE from Berger et al. [54]). The last four models are smoother than first two, while $\text{GARG}_{\text{B120k}}$ contains noise due to simulated acquisition. The evaluation points are also roughly selected by hand on the geometrically meaningful position illustrated in Fig. 3a: the center of the eye. The models do not share the same samples, so the evaluation position may change a bit, as well as the composition of the neighborhood for descriptor computation. Scale-space sampling and range are the same as in Fig. 3, with log basis $= 1.05$ and scale range $[1, s_3]$.

Output convolution profiles are shown in Fig. 6, and numerical values are available in additional materials. Among all the configurations, error of the estimation are in a range of $[-2, 2]$ in logarithmic scale-space, while the range of possibilities was $[-80, 80]$. The expected positions of the maximum are shown as vertical dashed line in the graphs. Note that $\text{GARG}_{\text{SCALED}}$ is two times smaller than all the other objects, and this factor is well detected by the estimations. Finally, the strength of the maxima visible in all the profiles is a proof of the robustness of our approach, which is not altered by the variation of sampling and lack of information, even for $\text{GARG}_{\text{P10k}}$ which does not contain any feature at fine and medium scales. Regarding the robustness to noise, the procedure used to fit this descriptor from Guennebaud and Gross [52] has been already proven to be robust to noise, and used with success to reconstruct surfaces from noisy point clouds in Berger et al. [53].

1. Respectively the reference model and the sample 9.

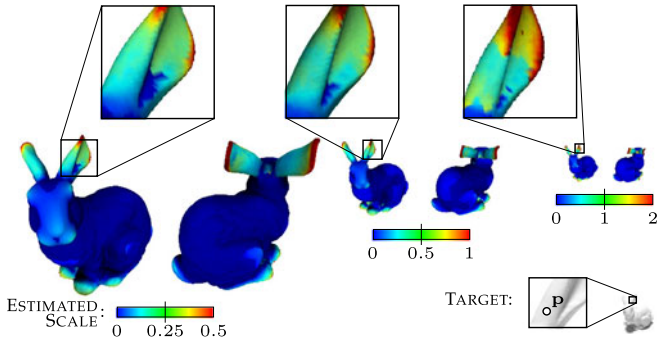


Fig. 7. Color-coded visualization of the estimated scale between all points in three different scaled versions of BUNNY, and $\mathbf{p}$. The target scale is always shown in green, under-estimations in blue, and over-estimations in red. Scale-space is sampled with a basis of 1.05, starting from 1 unit, with 100 samples (max. scale $\approx \frac{1}{2}$ bounding box size of the target model).

4.3.3 Robustness to Limited Overlapping and Outliers

Point-clouds and meshes acquired with different devices are perturbed by different types of noise (e.g., frequency, amplitude, pattern), and possibly by outliers. The robustness of our approach is directly related to the robustness of the GLS descriptor. However, this approach is theoretically designed to handle a limited amount of outliers, which can strongly influence the least-square minimization and lead to non-representative GLS descriptors. In practice, we observed that a realistic amount of outliers can be present in the data without perturbing the descriptor computation (see results in Fig. 9). Another issue is related to the overlap between two models: if one of the two describes only a portion of the surface, the profiles associated to two corresponding points could present partially different profiles. In this case, only the compatible parts of the descriptor will influence the convolution output.

Given the amount of overlap, the size of the profiles to be compared should be carefully chosen. For example, in an automatic scenario, the profile parameters could be refined after an initial relative scale estimation step.

4.3.4 Spatial Smoothness

An important aspect of our approach is that it allows to retrieve the right scale between two points which are not exactly in the same position on the geometry, i.e., the scale estimations vary smoothly on the surface. This spatial smoothness is illustrated in Fig. 7, where we show the estimated scales between a point $\mathbf{p}$, and all the points on three scaled version of the BUNNY.

There are two interesting behaviors we would like to emphasis in this example. First, the right scale is estimated on a subpart of the ear of the BUNNY, which mean that we do not need to pick exactly the same positions on the geometry to get a correct estimation. On the other hand, this stable area is not too big, and wrong scales are estimated on the head or the ear extremity. Second, this behavior is stable over relative scale variations without any change of the scale-space range or sampling. Our semi-automatic

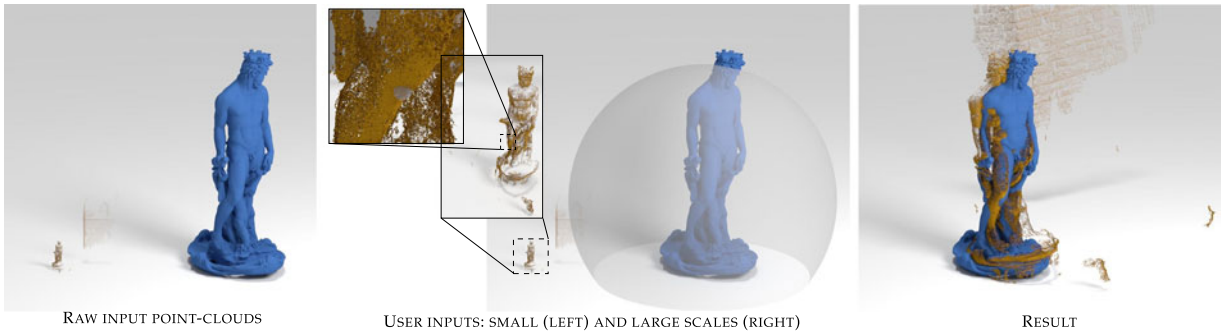


Fig. 8. Overview of our interactive registration tool, which uses a single pair of points specified by the user to estimate both the relative scale and the local frame between two acquired models, here $\text{BIANCONE}_{\text{SCAN}}$ (blue, 2 M points) and $\text{BIANCONE}_{\text{PMVS}}$ (gold, 1 M points). Neighbourhood collection, descriptor computation, and alignment takes less than 20 seconds (CPU using one single core) without pre-processing.

matching approach (see Section 5.1) takes advantage of this to allow the user not to be too accurate in selecting correspondences between models, and this is used also by the automatic approach to select starting *seeds* (see Section 5.2). Note that we get similar scale estimations for both ears of the BUNNY, because of the strong symmetry of the model. Ears can however been disambiguated if the scale interval is big enough, using details at large scales. Indeed, the magnitude of the maximum extracted in Eq. (6) encodes the similarity between points, and a bigger value means a better match.

5 MULTI-MODAL REGISTRATION

Registering 3D objects is a common step in the acquisition and digital modeling of physical objects. When dealing with complex real scene or objects, user intervention is most of the time necessary to guide processes. A typical scenario is to let the user setting some correspondences between two models and then align them. Automatic procedures can also be envisaged, especially when the data exhibit comparable size, amount of noise and sampling, and sufficient overlapping. As demonstrated in previous sections, multi-modal data usually break these conditions, and for that reason cannot be registered by such techniques. Another important issue is the increasing number of acquired data today accessible, which requires more and more efficient registration techniques, designed either as simple and fast supervised systems or automatic procedures.

We have implemented our point-wise scale estimation and comparison estimator in two concrete and efficient registration systems, designed to handle point-clouds composed by millions of points and corrupted by acquisition artefacts. To prove the versatility and robustness of our approach, we have used them to register both synthetic and acquired objects, and compared our results to existing techniques on our data. Our systems are designed as follow:

- *A semi-automatic approach* to interactively register models with a simple interaction, and without any pre-processing or subsampling.
- *An automatic approach* based on a standard RANSAC scheme and using our scale estimation to filter invalid configurations.

Both approaches are designed to be available independently, but one could use them processing pipeline where most of the data are processed automatically, and problematic cases fixed by user intervention.

5.1 Semi-Automatic Registration

In this approach we propose to use minimal user intervention to retrieve the relative scale between two acquired models, at interactive rate, without pre-computation or subsampling. The overall pipeline of our approach is illustrated in Fig. 8, and requires two inputs from the user: a scale range, and a pair of roughly corresponding points between the objects. The key idea is to let the user set the parameters that have a strong influence on the process, but are easy to set for a human being. Then, the relative transformation between the two models (scale, rotation and translation) is estimated automatically in interactive time. We would like to emphasize that this approach has been designed as a proof of concept, and requires small implementation efforts. It uses simple heuristics and neither complex optimization nor advanced hardware. Despite this simple design, it has been proven more efficient than existing tools, and applied successfully on real datasets.

5.1.1 *Interactive Scale Estimation and Matching*

According to Section 4.3, the scale range and sampling parameters could have a strong influence on the accuracy of the scale estimation. We let the user specify them, simply by picking points on the models and adjusting the profiles range by changing the global minimum scale, the logarithmic base and the number of scale samples (the maximum scale is automatically computed from these values). The changes of the parameters is visually shown by indicating the volume (or better to say, the sphere of influence) that will be taken into account for the profile calculation. The user can also take into account the amount of overlap between the two models and adjust the profile to better fit the common information between them.

Then, the user can select a pair of corresponding points and the system interactively finds the relative scale, and align the two models. The selected points can be edited and the result updated. Note that the user does not need to be very accurate in the point selection, thanks to the spatial smoothness of the scale estimation.

According to Section 4.2, we estimate the relative scale between the two selected points as a maximum of Δ_σ, while the translation between both models is found by simply aligning the two corresponding points. The rotation between the two models is retrieved by estimating a local frames for each selected point, and aligning them. The local frames are

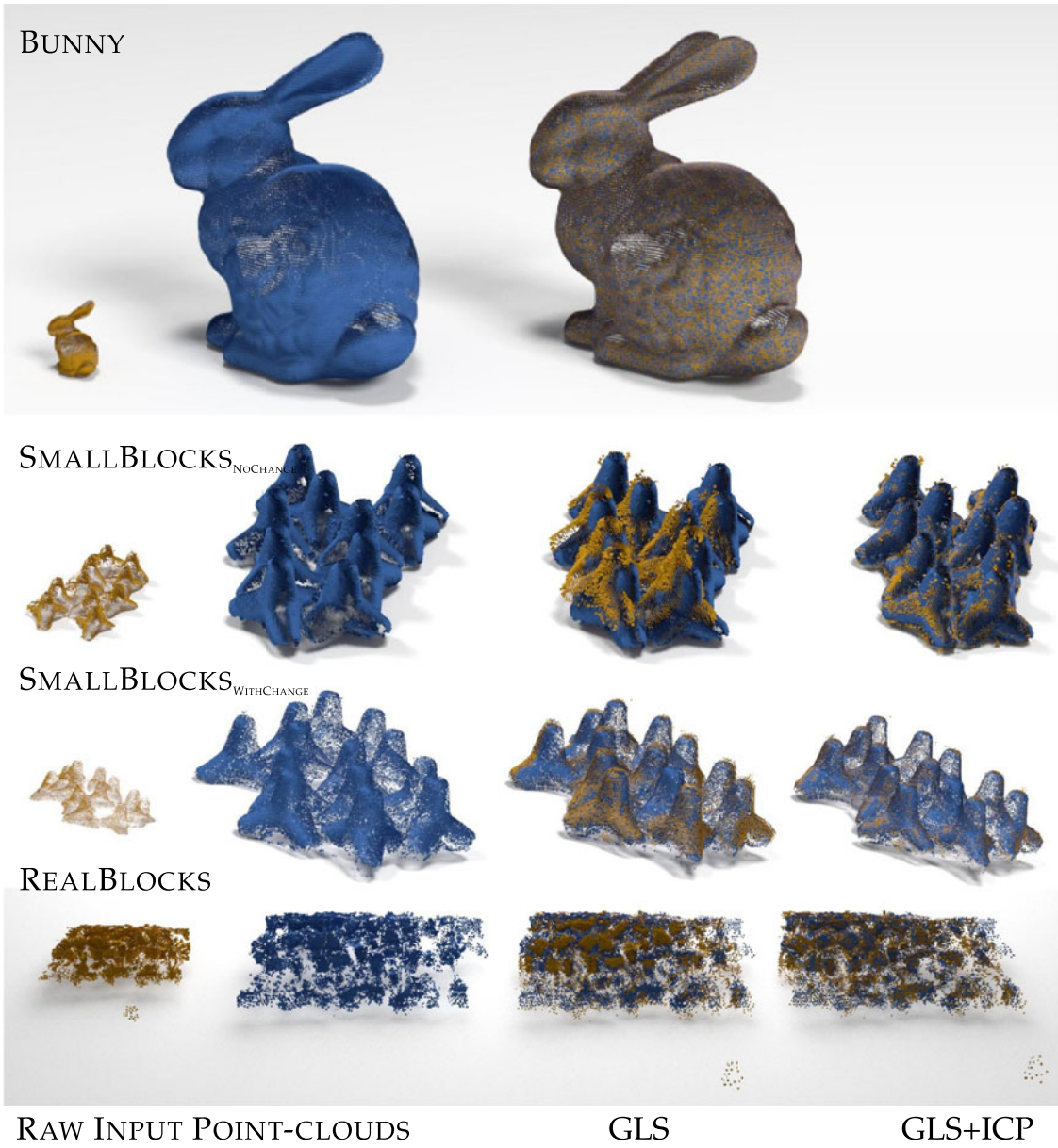
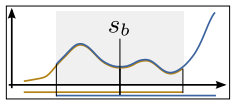


Fig. 9. Dataset used for scale-estimation comparison, from [28].

computed at a given scale as $\left(\boldsymbol{\eta},\quad \mathbf{d}_{\mathbf{k}_1},\quad \boldsymbol{\eta}\times\mathbf{d}_{\mathbf{k}_1}\right)$ where $\times$ is the cross product between two vectors, and $\mathbf{d}_{\mathbf{k}_1}$ the principal curvature direction computed by spatially differentiating the GLS descriptor [55].

A key aspect here is to choose the right scale to estimate the local basis: small scales are potentially noisy, and at very large scale (e.g., when considering the entire object) the principal curvature direction can be poorly descriptive. In practice, we first estimate the scale between two points, and then use the scale s_b in the middle of the overlapping interval to compute the local basis (see inset below). This heuristic worked for our tests, and can be easily tuned or interactively changed by the user.



5.1.2 Comparisons

We used Patate [55] to compute, match and differentiate the GLS descriptors. Neighborhoods are collected using a KdTree built on the full-resolution cloud, and computations were made using a single core of a Xeon CPU (3.00 GHz).

We compared our approach with the manual scale estimation and registration tool available in Meshlab [56] (see attached video). In this system, users have to set multiple pairs of corresponding points between two models, and a PCA is applied to retrieve the similarity matrix between them. Especially when scaling is taken into account, at least 4-5 correspondences pair must be chosen, and they should be well distributed in the context of the overlap between models. This operation can be quite time-consuming, especially when models are noisy or incomplete.

Table 2 shows a comparison of the performances on three datasets from Corsini et al. [27]. Both manual and semi-automatic approaches are able to provide accurate registration. However, our approach permits to complete the operation in a shorter time (including correspondence

TABLE 2
Timings of Our Semi-Automatic Scale Estimation and a Manual Approach Implemented in Meshlab

Input Data			Semi-automatic (our)			Manual	
Input	Reference	Target Scale	Nb Scales	Time (sec)	Est. scale	Time (sec)	Est. scale
$\text{GARG}_{\text{SCALED}}$ (1.5M vert.)	$\text{GARG}_{\text{B500k}}$ (500k vert.)	2.0	20	1.07 0.04	2.08	90	2.1
$\text{SPOUSE}_{\text{PMVS}}$ (1.4M vert.)	$\text{SPOUSE}_{\text{SCAN}}$ (4M vert.)	4.25	70	7.8 15.5	4.32	285	4.22
$\text{BIANCONE}_{\text{PMVS}}$ (970k vert.)	$\text{BIANCONE}_{\text{SCAN}}$ (2M vert.)	8.20	100	6.87 6.08	7.83	240	8.13

Our approach requires to compute the GLS descriptor around points on the reference and the input model (respectively first and second line in the Time column). The time to set the points was between 15 and 20 seconds in our experiments (the GLS of the first picked point is computed while choosing the second one), by using a simple trackball and a picking system. The manual approach requires to inspect and set accurately 4-5 pairs of points between the two models. In both cases the time required to estimated the scale is negligible (a few milliseconds).

setting) and requires less accuracy and effort by the user. Note that in the three examples the larger scale used to compute the GLS descriptor includes the whole input clouds. Hence, most of the computation time is in practice spend on neighborhood queries, which could be improved using multi-resolution schemes.

5.2 Automatic Registration

Depending on the application context, it might be necessary to automatically register two point clouds, and avoid user intervention. In that case, automatic registration systems need to explore the transformation space (scaling and rigid transformation). In order to reduce the search space, one may use hints from the models geometry. A standard approach is to detect representative points within the clouds, called *seeds* in the following, and find correspondences between them. In our semi-automatic approach, this critical step is done by the user using context-specific knowledge. A wide range of automatic registration techniques have been proposed in the past to explore the transformation space using points correspondences, e.g., RANSAC [57] and evolutionary game theoretic matching [58]. We choose to design our automatic registration method as a simple RANSAC scheme, and demonstrate its efficiency on real-world dataset. More involved approaches can be derived from this for specific application context.

In order to reduce the amount of points and reduce the computational load, we start by sub-sampling the two clouds with a variant of the Constrained Poisson Disk approach [54]. This approach outputs a homogeneous distribution of samples while preserving the details (we used 200 k samples in our experiments). We then pre-compute the GLS descriptors on these clouds. The automatic registration procedure is divided in two steps: first the *seeds* extraction and matching, then the transformation space exploration using RANSAC.

5.2.1 Seeds Selection and Matching

In Section 4.3, we showed that our scale estimation is robust to variations of sampling density, noise, and more importantly that it evolves smoothly on objects. This property is really important since it allows us to compute a good relative scaling without requiring *accurate* corresponding points. Even if the corresponding points are not taken in in the exact relative position on the two models, the scale estimation can be accurately performed. Nevertheless, according to [24], using points which are too close one to another during the registration increases the chance to estimate a noisy transformation. Special care must then be taken to avoid having all the *seeds* located at the same place on the model and ensure that they cover as much as possible the important features of the object. We ensure a minimal distance m_d between *seeds* by sampling a second time the models using [54], with a sampling density is defined w.r.t m_d (we used between 2 k and 3 k points for our experiments). The value of m_d can be calculated by taking into account the size of the bounding box of the object.

The resulting set defines the *seeds* used later to define correspondences. Instead of using a totally random approach, we decided to prioritize the *seeds* in order to increase the speed and accuracy of the registration procedure. We prioritize the *seeds* using the geometric variation ν proposed in [5], and defined as

$$\nu(\mathbf{p}, s) = w_\tau \left(\frac{\delta \tau_s}{\delta s}\right)^2 + w_{\boldsymbol{\eta}} \left(s \frac{\delta \boldsymbol{\eta}}{\delta s}\right)^2 + w_\kappa \left(s^2 \frac{\delta \kappa_s}{\delta s}\right)^2.$$

As in Equation (1) we used $w_\tau = w_{\boldsymbol{\eta}} = w_\kappa = 1$. The priority of a *seed* is computed as

$$priority(\mathbf{p}) = \frac{1}{\sum_{s_{min}}^{s_{max}} 1} \sum_{s=s_{min}}^{s_{max}} 1 - tanh(\alpha * \nu(\mathbf{p}, s)) \qquad (7)$$

with α defined as in Equation (4). The intuition behind this measure is to prefer points exhibiting variations in their GLS descriptor, producing a more distinctive signature for the scale estimation process.

Then, each *seed* $i_k \in I$ in the input model is matched to its three most similar points in the reference model using Eq. (5) with h computed using Equation (6). Both the estimated scale and the points priorities are attached to the generated pair. A set of three *pairs* is created for each *seed*. We note P the priority queue storing the pairs of points of the reference and input models, with the priority of a pair defined as the product of its points priority.

5.2.2 Finding Relative Scale and Registration

In a second stage, we use the priority queue of *pairs* generated by the previous step to find a the correct relative scale and an initial registration to be refined in a subsequent stage. We use the RANSAC scheme described in Algorithm 1. If no solution is found after a chosen number of iterations, another set of *seeds* I is computed on the input model, and the procedure starts again. For the sake of clarity, we described sub-procedures in Appendix.

TABLE 3
Comparison of Relative Scale Estimation for Several Methods, with Estimated Scale and Percentage Error

Dataset	Ground Truth	Standard Deviation	Mesh Resolution [59]	Keyscale [60]	Standard ICP [61]	Scale Ratio ICP [28]	GLS	GLS + ICP [1]
BUNNY	5.000	5.000	5.000	5.000	5.000	5.000	5.000	5.000
Small Blocks (no change)	2.364	4.855 (105.37%)	1.162 (50.85%)	1.400 (40.78%)	3.029 (28.13%)	2.502 (5.84%)	**2.430 (2.81%)**	**2.382 (1.01%)**
Small Blocks (with change)	2.424	3.833 (58.13%)	1.684 (30.53%)	2.250 (7.18%)	2.561 (5.65%)	2.543 (4.91%)	**2.525 (4.16%)**	**2.505 (3.34%)**
Real blocks	1.696	1.593 (6.07%)	1.607 (5.25%)	1.500 (11.56%)	1.767 (4.19%)	1.607 (5.25%)	**1.662 (3.01%)**	**1.674 (2.30%)**

The last two columns show the result of our method with or without ICP. The models and the table (except for the last two columns) are from [28]. Our results are shown in Fig. 9.

Algorithm 1. RANSAC Scheme used to Explore the Transformation Space using Pairs of *Seeds* from the Two Clouds

Data: PriorityQueue<Pairs >P,Q;
Result: TransformationMatrix M
while *IterationCount* <*ItMax* **do**
  triplet$_{ref}$ = `ExtractTriplet`(P);
  scale_ok = `scaleDiff`(triplet$_{ref}$) $<1 \pm e_s$;
  if *scale_ok* **then**
    M = `ComputeRigidTr`(triplet$_{ref}$);
    if (`RegistrationErr`($triplet_{ref}$, M) $<e_p$
    AND `NormalErr`($triplet_{ref}$, M) $<e_n$) **then**
      Q=P ;
      while !$Q.isempty()$ **do**
        $\mathbf{q}_{ref}$ = `ExtractFourthPair`($Q.pop()$);
        if `IsValid`($triplet_{ref}$,q_{ref}) **then**
          return
          `ComputeRigidTr`($triplet_{ref}$,q_{ref});
        end
      end
    end
  end
end
return *IdentityMatrix*

5.2.3 Parameters

All the results in the next section were obtained with a logarithmic base of 1.2. The minimum and maximum scale values used to compute the GLS profile were computed respectively as the average distance between the samples and as the diagonal of the bounding box, for each cloud independently. The other parameters were set as $m_d = 1\%$ of the diagonal of the bounding box, $e_p = 4.0$ units (the reference models were all in millimeters), $e_n = 20$ degrees, and $e_s = 0.2$. All the results were obtained using this single parameter set.

5.3 Comparisons

According to Section 2, while several recent works are facing the issue of scale estimation, only a few of them can actually be applied on real multi-modal data. This is mainly due restrictions on the data representation (triangulated surfaces, 3D scans) [26], [32], [33] or the strong sensitivity to noise. Other techniques couldn't be reproduced [35], since the code was not made available to the community.

First, we compared our approach with the work of Lin and colleagues [28], which is as far as we know the only approach robust enough to handle multimodal data. According to the authors, the approach is quite sensitive to the amount of overlap between the clouds. Nevertheless, we compared our approach to previous work on datasets proposed in [28], to test the accuracy in relative scale estimation among some state-of-the-art techniques. Results of our approach are shown in Fig. 9, and quantitative values in Table 3. Our method outperforms the other also before the use of ICP to refine alignment.

Since our method aims at dealing with challenging real datasets, we also tested our approach on models exhibiting strong differences in density, noise, and coverage (results shown in Fig. 10 and in the attached video). According to these results our approach is able to handle complex scenes. Processing time (see Fig. 10) is mainly influenced by the complexity of the scene and the amount of overlap.

We observed that our approach tended to fail for data with very low overlap, or when the geometry was scarcely representative (i.e., vast majority of flat surfaces). We would like to emphasize that any other descriptor-based approach may have similar limitations.

6 Conclusion and Future Improvements

We presented a method for the registration and relative scale estimation of multi-modal geometric data. Our method uses a descriptor based on Growing Least Squares, which is able to characterize both local geometric details and global shape properties on 3D objects. We introduced a new operator to compare two GLS descriptors at arbitrary scales, compute their similarity in scale-space and estimate a relative scale factor between them. Thanks to its robustness, this operator is a good candidate to compare data with variable amount of details, noise, and sampling. In addition, our approach is easy to implement, and fast to evaluate even on point-clouds composed of millions of points. We evaluated it on data corrupted by acquisition artifacts, and shown its stability and relevance for the study of multi-model 3D data.

We built, upon this point-wise scale estimation, two practical frameworks to register multi-modal data either using a user-assisted or an automatic approach. In both cases, we used them to successfully register models with varying outliers, noise, sampling density, and holes. These challenging meshes and point clouds had been generated either by CG artists, or acquired using laser scanners, LIDAR, multi-view stereo and spherical photogrammetry. In all cases our approaches outperformed existing techniques, and in some case they were the only one able to find a solution.

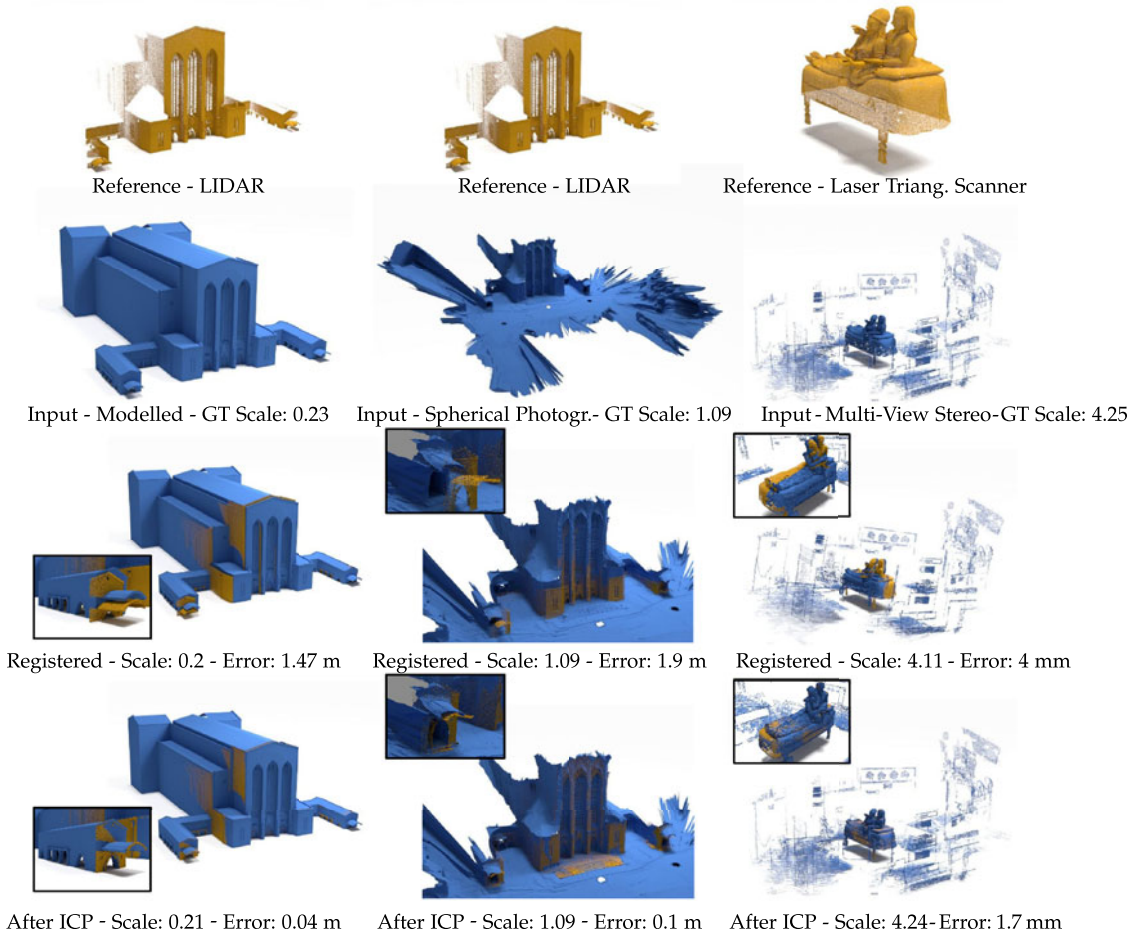


Fig. 10. Results of the method on real datasets. Left Column: LIDAR versus Modelled, processing time for first registration 170 secs. Middle Column: LIDAR versus Spherical photogrammetry, 615 secs. Right Column: 3D Scanning versus Multi-View Stereo, 412 secs.

It is important to emphasize that our approach is easy to implement, and can be easily diffused (using open-source library and softwares), reproduced or implemented in existing pipelines. A typical application example is to use it in systems gathering heterogeneous 3D contributions about real objects, and automatically register them in a common reference system and populate a virtual environment.

Thanks to this approach, the registration of 3D multi-modal data can now be easily obtained. However, there are still interesting and challenging questions we would like to study in future work. First, the detections of pertinent structures on acquired point-clouds is still a complex task requiring further study. It is a critical step used in many processing scenarios, like the *seed* selection in our context. In [5], authors present a continuous measure to detect pertinent structures in scale-space. In practice, it is not straightforward to use this measure to drive adaptive sampling techniques and extract pertinent points. A interesting research direction could be to improve such pertinent point extraction, which could significantly improve the performance and the robustness of the registration.

Moreover, this can be crucial also to handle the issue of partial overlapping between models. Regarding this problem, more advanced strategies to refine the profile parameters could lead to a higher robustness of the method.

Finally, another interesting direction is the improvement of both assisted and automatic approaches, either by designing dedicated user-interfaces, or by optimizing the matching algorithms and implement them on GPU.

APPENDIX

The relative scale estimation and geometric registration procedure shown in Section 5.2 is described step-by-step in this Appendix. The *PriorityQueue* elements are *Pairs*. Each Pair is described by: a point on the reference model i_{ref}, a point on the input model i_{inp}, an associated scale factor s, and a priority value p. The function *ExtractTriplet* extracts three pairs from the PriorityQueue. The probability to be extracted is correlated to the priority value associated to each pair. The first check on the triplet of pairs is the similarity of estimated scale factor (Algorithm 2). If the errors are above the defined threshold (respectively e_p and e_n), the triplet is discarded. Then, the Priority Queue P is copied on another Priority Queue Q, and every point q_{ref} of Q is checked to validate the triplet (Algorithm 5). The fourth pair selected must have a distance bigger then m_d with all the pairs of the reference triplet, and the triplet+fourth pair Registration fulfils the thresholds e_p and e_n. If the returned scale difference is above the e_s threshold, the triplet is discarded. Otherwise, standard PCA algorithm is applied to calculate the Registration Matrix M that best aligns the pairs of the Triplet. Then the Registration Error (Algorithm 3) and the Normals Error (Algorithm 4) are calculated in the standard way. If a fourth pair that validates the Triplet is

found, the Registration Matrix obtained from the four pairs is returned as the final result.

Algorithm 2. The *scaleDiff* Function

Data: Triplet *triplet*
Result: Value *scaleDifference*

avgScale=(triplet.s_1+triplet.s_2+triplet.s_3)/3;
scaleDifference=(abs((triplet.s_1-avgScale))
+ abs((triplet.s_2-avgScale)) + abs((triplet.s_3-avgScale)))/3;
return *scaleDifference*

Algorithm 3. The *RegistrationErr* Function

Data: Pairs *pairs*, Registration Matrix M
Result: Value *regError*

regError=(abs($pairs_1.i_{ref}$- M X $pairs_1.i_{inp}$) + abs($pairs_2.i_{ref}$- M X $pairs_2.i_{inp}$) + $\cdots$ + abs($pairs_{k1}.i_{ref}$- M X $pairs_k.i_{inp}$))/k ;
return *regError*

Algorithm 4. The *NormalErr* Function. n_i is the normal associated to the *i*th point of the pair. *ang()* calculates the angle between the two vectors.

Data: Pairs *pairs*, Registration Matrix M
Result: Value *normError*

normError=(*ang*($pairs_1.n_{ref}$,MX $pairs_1.n_{inp}$)+*ang*($pairs_2.n_{ref}$,MX $pairs_2.n_{inp}$)+ $\cdots$ +*ang*($pairs_k.n_{ref}$,M X $pairs_k.n_{inp}$))/k ;
return *normError*

Algorithm 5. The *IsValid* Function

Data: Triplet *triplet*, Pair *q*
Result: Bool *IsValid*
if `minDist`(*triplet*, *q*) $>m_d$ **then**
  return *false*;
end
M_1 = `ComputeRigidTr`(<*triplet*, *q*>);
if `RegistrationErr`(<*triplet*, *q*, M_1>) $<e_p$
AND `NormalErr`(<*triplet*, *q*, M_1>) $<e_n$ **then**
  return *true*;
end
else
  return *false*;
end

ACKNOWLEDGMENTS

The research leading to these results was partially funded by EU FP7 project ICT FET Harvest4D (http://www.harvest4d.org/, G.A. no. 323567), ERC Starting Grant SmartGeometry (StG-2013-335373), and ANR Mapstyle project (ANR-12-COORD-0025). The authors wish to thank Paolo Cignoni, Niloy Mitra and David Vanderhaghe for their support and advices, as well as Pascal Barla, Gael Guennebaud, Aron Monszpart and Moos Hueting for comments and discussions. Part of this work has been done at University College London, Department of Computer Science, London.

REFERENCES

[1] S. Du, N. Zheng, L. Xiong, S. Ying, and J. Xue, "Scaling iterative closest point algorithm for registration of m-D point sets," *J. Vis. Commun. Image Represent.*, vol. 21, nos. 5–6, pp. 442–452, Jul. 2010.
[2] K. Parmar and R. Kher, "A comparative analysis of multimodality medical image fusion methods," in *Proc. Modelling Symp.*, May 2012, pp. 93–97.
[3] M. Bhattacharya and A. Das, "Multimodality medical image registration and fusion techniques using mutual information and genetic algorithm-based approaches," *Softw. Tools Algorithms Biological Syst.*, vol. 696, pp. 441–449, 2011.
[4] I. Reducindo, E. Arce-Santana, D. Campos-Delgado, and A. Alba, "Evaluation of multimodal medical image registration based on particle filter," in *Proc. Elect. Eng. Comput. Sci. Autom. Control*, Sep. 2010, pp. 406–411.
[5] N. Mellado, G. Guennebaud, P. Barla, P. Reuter, and C. Schlick, "Growing least squares for the analysis of manifolds in scale-space," *Comp. Graph. Forum*, vol. 31, no. 5, pp. 1691–1701, Aug. 2012.
[6] J. Salvi, C. Matabosch, D. Fofi, and J. Forest, "A review of recent range image registration methods with accuracy evaluation," *Image Vis. Comput.*, vol. 25, no. 5, pp. 578–596, May 2007.
[7] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or, "A survey on shape correspondence," *Comput. Graph. Forum*, vol. 30, no. 6, pp. 1681–1707, 2011.
[8] G. Tam, Z.-Q. Cheng, Y.-K. Lai, F. Langbein, Y. Liu, D. Marshall, R. Martin, X.-F. Sun, and P. Rosin, "Registration of 3D point clouds and meshes: A survey from rigid to nonrigid," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 7, pp. 1199–1217, Jul. 2013.
[9] H. Pottmann, S. Leopoldseder, and M. Hofer, "Registration without ICP," *Comput. Vis. Image Understanding*, vol. 95, no. 1, pp. 54–71, 2004.
[10] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. 3rd Int. Conf. 3D Digital Imaging Model.*, Jun. 2001, pp. 145–152.
[11] N. J. Mitra, N. Gelfand, H. Pottmann, and L. Guibas, "Registration of point cloud data from a geometric optimization perspective," in *Proc. Symp. Geom. Process.*, 2004, pp. 22–31.
[12] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," in *Proc. Symp. Geom. Process.*, 2013, vol. 32, no. 5, pp. 1–11.
[13] P. Heider, A. Pierre-Pierre, R. Li, and C. Grimm, "Local shape descriptors, a survey and evaluation," in *Proc. 4th Eurograph. Conf. 3D Object Retrieval*, 2011, pp. 49–56.
[14] A. Johnson, "Spin-images: A representation for 3-d surface matching," Ph.D. dissertation, Robotics Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Aug. 1997.
[15] E. Kalogerakis, D. Nowrouzezahrai, P. Simari, and K. Singh, "Extracting lines of curvature from noisy point clouds," *Comput. Aided Des.*, vol. 41, no. 4, pp. 282–292, Apr. 2009.
[16] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 3212–3217.
[17] X. Li and I. Guskov, "Multi-scale features for approximate alignment of point-based surfaces," in *Proc. Symp. Geom. Process.*, 2005, p. 217.
[18] L. Skelly and S. Sclaroff, "Improved feature descriptors for 3-D surface matching," in *Proc. SPIE Conf. Two- 3-Dimensional Methods Inspection Metrol.*, vol. 6762, pp. 67620A–67620A–12, 2007.
[19] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust global registration," in *Proc. 3rd Eurograph. Symp. Geom. Process.*, 2005, p. 197.
[20] A. Makadia, A. I. Patterson, and K. Daniilidis, "Fully automatic registration of 3D point clouds," in *Proc. Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2006, pp. 1297–1304.
[21] H. Pottmann, Q.-X. Huang, Y.-L. Yang, and S.-M. Hu, "Geometry and convergence analysis of algorithms for registration of 3D shapes," *Int. J. Comput. Vis.*, vol. 67, no. 3, pp. 277–296, 2006.
[22] S. Krishnan, P. Y. Lee, J. B. Moore, and S. Venkatasubramanian, "Global registration of multiple 3D point sets via optimization-on-a-manifold," in *Proc. Symp. Geomet. Process.*, 2005, pp. 187–196.
[23] F. Bonarrigo and A. Signoroni, "An enhanced 'optimization-on-a-manifold' framework for global registration of 3D range data," in *Proc. Int. Conf. 3D Imag., Model., Process., Vis. Transmiss.*, 2011, pp. 350–357.
[24] D. Aiger, N. J. Mitra, and D. Cohen-Or, "4-points congruent sets for robust pairwise surface registration," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 85:1–85:10, Aug. 2008.

[25] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4PCS fast global pointcloud registration via smart indexing," *Comput. Graph. Forum*, vol. 33, no. 5, pp. 205–215, 2014.
[26] E. Rodola', A. Albarelli, F. Bergamasco, and A. Torsello, "A scale independent selection process for 3D object recognition in cluttered scenes," *Int. J. Comput. Vis.*, vol. 102, nos. 1–3, pp. 129–145, 2013.
[27] M. Corsini, M. Dellepiane, F. Ganovelli, R. Gherardi, A. Fusiello, and R. Scopigno, " Fully automatic registration of image sets on approximate geometry," *Int. J. Comput. Vis.*, vol. 102, nos. 1–3, pp. 91–111, 2013.
[28] B. Lin, T. Tamaki, F. Zhao, B. Raytchev, K. Kaneda, and K. Ichii, " Scale alignment of 3D point clouds with different scales," *Mach. Vis. Appl.*, vol. 25, no. 8, pp. 1989–2002, 2014.
[29] R. Pintus, E. Gobbetti, and R. Combet, "Fast and robust semi-automatic registration of photographs to 3D geometry," in *Proc. 12th Int. Conf. Virtual Reality, Archaeol. Cultural Heritage*, 2011, pp. 9–16.
[30] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys, "3D model matching with viewpoint-invariant patches (vip)," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.
[31] H. Kim and A. Hilton, "Evaluation of 3D feature descriptors for multi-modal data registration," in *Proc. Int. Conf. 3D Vis.*, June 2013, pp. 119–126.
[32] S. Lee, M. Park, and K. Lee, " Full 3D surface reconstruction of partial scan data with noise and different levels of scale," *J. Mech. Sci. Technol.*, vol. 28, no. 8, pp. 3171–3180, 2014.
[33] L. Quan and K. Tang, "Polynomial local shape descriptor on interest points for 3D part-in-whole matching," *Comput.-Aided Des.*, vol. 59, pp. 119–139, 2015.
[34] D. Cohen-Steiner, P. Alliez, and M. Desbrun, "Variational shape approximation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 905–914, Aug. 2004.
[35] R. Raguram and J.-M. Frahm, "RECON: Scale-adaptive robust estimation via residual consensus," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1299–1306.
[36] C. Wang, M. M. Bronstein, A. M. Bronstein, and N. Paragios, "Discrete minimum distortion correspondence problems for non-rigid shape matching," in *Proc. 3rd Int. Conf. Scale Space Variational Methods Comput. Vis.*, 2012, pp. 580–591.
[37] D. Raviv, A. M. Bronstein, M. M. Bronstein, R. Kimmel, and N. Sochen, "Affine-invariant geodesic geometry of deformable 3D shapes," *Comput. Graph.*, vol. 35, no. 3, pp. 692–697, 2011.
[38] R. M. Rustamov, "Laplace-Beltrami eigenfunctions for deformation invariant shape representation," in *Proc. 5th Eurograph. Symp. Geom. Process.*, 2007, pp. 225–233.
[39] A. Dubrovina and R. Kimmel, "Matching shapes by eigendecomposition of the laplace-beltrami operator," in *Proc. Int. Symp. 3D Data Process. , Vis. Transmiss.*, 2010, vol. 2, no. 3.
[40] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," in *Proc. Symp. Geom Process.*, 2009, pp. 1383–1392.
[41] M. M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 1704–1711.
[42] K. Crane, C. Weischedel, and M. Wardetzky, "Geodesics in heat: A new approach to computing distance based on heat flow," *ACM Trans. Graph.*, vol. 32, no. 5, pp. 152:1–152:11, Oct. 2013.
[43] G. Patané and M. Spagnuolo, "Heat diffusion kernel and distance on surface meshes and point sets," *Comput. Graph.*, vol. 37, no. 6, pp. 676–686, 2013.
[44] H. Fadaifard, G. Wolberg, and R. Haralick, "Multiscale 3D feature extraction and matching with an application to 3D face recognition," *Graph. Models*, vol. 75, no. 4, pp. 157–176, 2013.
[45] B. Romeny, *Front-End Vision and Multi-Scale Image Analysis: Multi-scale Computer Vision Theory and Applications, written in Mathematica*. 1st ed. New York, NY, USA: Springer, 2009.
[46] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. Int. Conf. Comput. Vis.*., 1999, p. 1150.
[47] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. 15th Int. Conf. Multimedia*, 2007, pp. 357–360.
[48] G. Flitton, T. Breckon, and N. Megherbi Bouallagu, "Object recognition using 3D sift in complex CT volumes," in *Proc. British Mach. Vis. Conf.*, 2010, pp. 11.1–11.12.
[49] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud, "Surface feature detection and description with applications to mesh matching," in *Proc. IEEE Conf. Comput. Vis. Patern Recog.*, 2009, pp. 373–380.
[50] C. Maes, T. Fabry, J. Keustermans, D. Smeets, P. Suetens, and D. Vandermeulen, "Feature detection on 3D face surfaces for pose normalisation and recognition," in *Proc. 4th Int. Conf. Biometrics: Theory Appl. Syst.*, 2010, pp. 1–6.
[51] T. Darom and Y. Keller, "Scale-invariant features for 3-D mesh models," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2758–2769, May 2012.
[52] G. Guennebaud and M. Gross, "Algebraic point set surfaces," *ACM Trans. Graph.*, vol. 26, no. 3, p. 23, Jul. 2007.
[53] M. Berger, J. A. Levine, L. G. Nonato, G. Taubin, and C. T. Silva, "A benchmark for surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 2, pp. 20:1–20:17, Apr. 2013.
[54] M. Corsini, P. Cignoni, and R. Scopigno, "Efficient and flexible sampling with blue noise properties of triangular meshes," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 6, pp. 914–924, Jun. 2012.
[55] N. Mellado, G. Ciaudo, G. Guennebaud, and P. Barla. (2013). Patate lib [Online]. Available: http://patate.gforge.inria.fr/
[56] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "Meshlab: An open-source mesh processing tool," in *Proc. 6th Eurograph. Italian Chapter Conf.*, 2008, p. 129.
[57] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[58] A. Albarelli, E. Rodola, and A. Torsello, "Loosely distinctive features for robust surface alignment," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 519–532.
[59] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.
[60] T. Tamaki, S. Tanigawa, Y. Ueno, B. Raytchev, and K. Kaneda, "Scale matching of 3D point clouds by finding keyscales with spin images," in *Proc. 20th Int. Conf. Pattern Recog.*, 2010, pp. 3480–3483.
[61] P. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

Nicolas Mellado received the PhD degree in computer science from the University of Bordeaux, France, December 2012. He is a post-doctoral researcher at the IRIT, University of Toulouse. His research interests include point cloud processing, multiscale analysis, and registration.

Matteo Dellepiane received an advanced degree in Telecommunication Engineering (Laurea) from the University of Genova, in 2002, and the PhD degree in Information Engineering from the University of Pisa, in 2009. He is a researcher at the CNR-ISTI. His research interests include 3D scanning, digital archeology, color acquisition, and visualization on 3D models and web visualization of 3D models.

Roberto Scopigno graduated in computer science at the University of Pisa, in 1984. He is a research director at the CNR-ISTI and leads the Visual Computing Lab. He is engaged in research projects concerned with 3D scanning, surface reconstruction, multiresolution, scientific visualization, and cultural heritage. He is currently editor-in-chief of the *ACM J. on Computing and Cultural Heritage*.