A Qualitative Review on 3D Coarse Registration Methods

YAGO DÍEZ, FERRAN ROURE, XAVIER LLADÓ, and JOAQUIM SALVI, University of Girona

3D registration or matching is a crucial step in 3D model reconstruction. Registration applications span along a variety of research fields, including computational geometry, computer vision, and geometric modeling. This variety of applications produces many diverse approaches to the problem but at the same time yields divergent notations and a lack of standardized algorithms and guidelines to classify existing methods. In this article, we review the state of the art of the 3D rigid registration topic (focused on Coarse Matching) and offer qualitative comparison between the most relevant approaches. Furthermore, we propose a pipeline to classify the existing methods and define a standard formal notation, offering a global point of view of the literature.

Our discussion, based on the results presented in the analyzed papers, shows how, although certain aspects of the registration process still need to be tested further in real application situations, the registration pipeline as a whole has progressed steadily. As a result of this progress in all registration aspects, it is now possible to put together algorithms that are able to tackle new and challenging problems with unprecedented data sizes and meeting strict precision criteria.

Categories and Subject Descriptors: I.3.5 [**Curve, Surface, Solid, and Object Representations**]: 3D Registration

General Terms: Algorithms, Performance, Standardization

Additional Key Words and Phrases: 3D registration, coarse matching, surface alignment, point cloud matching, point descriptors, computational geometry

ACM Reference Format:
Yago Díez, Ferran Roure, Xavier Lladó, and Joaquim Salvi. 2015. A qualitative review on 3D coarse registration methods. ACM Comput. Surv. 47, 3, Article 45 (February 2015), 36 pages.
DOI: http://dx.doi.org/10.1145/2692160

1. INTRODUCTION

3D registration[1] represents a fundamental problem in a variety of areas, such as medical imaging, heritage reconstruction, shape retrieval, and industrial applications. Specific issues include alignment of temporal 3D images for lesion monitoring, modeling of structures, and the reconstruction of an object giving several views, or the bin picking problem.

[1] Note that we understand the words *registration*, *matching*, and *alignment* as synonyms, and we use them interchangeably throughout the article.

This work has been supported by the FP7-ICT-2011-7 project PANDORA-Persistent Autonomy through Learning, Adaptation, Observation and Re-planning (Ref 288273) funded by the European Commission and the project RAIMON-Autonomous Underwater Robot for Marine Fish Farms Inspection and Monitoring (Ref CTM2011-29691-C02-02) funded by the Ministry of Economy and Competitiveness of the Spanish government. Ferran Roure is supported by an FPI scholarship associated with the RAIMON project.

Authors' addresses: Y. Díez, F. Roure, X. Lladó, and J. Salvi; emails: {yago, froure, llado, qsalvi}@eia.udg.edu.

DOI: http://dx.doi.org/10.1145/2692160

Registration methods work with different types of input data that we categorize as (1) synthetic data (totally computer made), (2) processed data (filtered and modified scanned data), and (3) real data (scanned data without any modification). Current methods of data acquisition (scanners, structured light, etc.) are able to provide huge amounts of data corresponding to precise reconstructions. Depth cameras are an emergent acquisition technology, a representative example being Microsoft Kinect [Lui et al. 2012], which is gaining in popularity because it provides good performance at reasonable prices. However, in these cases, the raw depth data must be processed to obtain the geometric primitives used in registration algorithms [Khoshelham and Elberink 2012]. The scanned information obtained from any type of scanner can be handled using a variety of these geometric primitives. Point clouds are the simplest representation and are widely used in the literature. However, many methods need more complex structures, such as triangular meshes.

The size of the input data used in most specific applications makes the development of efficient algorithms a key issue. For example, in object reconstruction, the most popular strategy is still to get many different views of the model and subsequently register them onto a common coordinate system. 3D registration allows for full model reconstruction; however, if a high degree of precision is required, such as a huge number of points in the cloud, the process requires highly efficient methods to achieve registration in a reasonable amount of time. Although matching algorithms have seen many improvements over recent years, there is still no algorithm that can be considered standard in the sense that it can be used reliably in all situations and with the desired data sizes.

Due to the high number of application fields of registration, different scientific communities produce contributions related to it. These communities include computer graphics (Eurographics conference,[2] SIGGRAPH conference[3]), computational geometry (SoCG conference,[4] *Journal of Computational Geometry*[5]), and computer vision (*Pattern Recognition* journal,[6] *International Journal of Computer Vision*,[7] *IEEE Transactions on Pattern Analysis and Machine Intelligence*[8]), to name a few. This dispersion of contributions makes the organization of information more difficult. The main problems are the lack of a common notation, the diversity of interests when approaching similar problems, and the lack of common evaluation criteria. In this article, we propose a pipelined classification for the methods involved in the registration process. Our aims are to relate divergent notations addressing similar issues, review the most popular methods for each application area, and classify them according to the aspects of their matching processes.

Although we focus on 3D rigid registration methods, many other registration-related problems exist, such as nonrigid alignment [Huang et al. 2008; Kumar et al. 2001], shape morphing [Alexa 2002], deformation transfer, self-similarity detection, or time-varying surface reconstruction. For further details on these areas, we recommend two qualitative reviews: Van Kaick et al. [2011] and Tam et al. [2013].

The rest of this article is structured as follows. Section 2 presents an overview of the state of the art as well as a generic pipelined classification of the different steps used in point cloud registration. Section 3 defines the problem and presents the formal notation

[2] http://www.eg.org/.

[3] http://www.siggraph.org/.

[4] http://www.uniriotec.br/∼socg2013/.

[5] http://jocg.org/index.php/jocg.

[6] http://www.journals.elsevier.com/pattern-recognition/.

[7] http://link.springer.com/journal/11263.

[8] http://www.computer.org/portal/web/tpami.

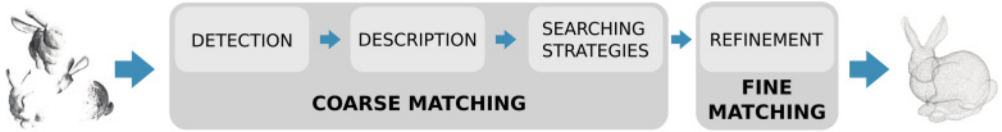

Fig. 1. Point registration pipeline.

used in this article. Sections 4, 5, 6, and 7 provide details on the four main steps defined on the aforementioned pipeline and describe the most relevant approaches in every step. In Section 8, we evaluate the methods studied according to practical criteria and discuss their performances. Finally, Section 9 focuses on the conclusions of this review.

2. OVERVIEW

The registration process consists of several steps. We propose a pipelined classification (Figure 1) to organize existing techniques in each step of the registration process. Methods are divided into two different categories, according to the initial relative position of the data: Coarse Matching and Fine Matching. *Coarse Matching*, on which we focus in this article, encompasses all techniques that return a rough initial alignment of the input point clouds placed anywhere, without any initial alignment. In the literature, we can find these kinds of methods under different terms such as *Rough* or *Coarse* alignment, or *Global* or *Crude* registration. On the other hand, *Fine Matching* includes methods that start from one such approximation and aim at finding a registration as accurate as possible. Coarse Matching can be further divided into three different steps: detection, description, and searching strategies. As we will see throughout this article, most approaches are focused on only one part of the pipeline. Most frequently, this part is the detection or description step. In most cases, the rest of the pipeline is completed using very basic methods or even brute force.

Although the distinction is clearly defined between Coarse and Fine Matching, some methods within Coarse Matching are difficult to categorize. There are methods encompassing different steps, such as detection and description or description and searching strategies.

Thus, our pipelined classification is structured as follows. First of all, a detection step is used to reduce the number of points being considered. It consists of detecting a certain number of key points that are prominent according to a specific criterion. The sizes of input data make the detection step necessary in many approaches to obtain computationally manageable datasets. The second step of the pipeline, description, consists of assigning values to the detected key points according to the properties of the shape around them. The functions that perform this are called *local shape descriptors*. Finally, searching strategies are used to find correspondences between points in the two point sets. A correspondence between two points from different point clouds reinforces the assumption that these two points will be the same in the final registered shape. Descriptor values are used to prioritize the best apparent correspondences. A minimum of three correspondences are needed to determine the coarse alignment in 3D. The goal in this case is to avoid exhaustive search of the whole correspondence space. This exhaustive search would lead to asymptotic costs of $O(n^6)$ for 3D registration (n being the number of points of the sets), because corresponding triplets that determine the movement are found by checking all point combinations from both shapes. After achieving coarse alignment, a Refinement step is applied. This step consists of using iterative methods to align the shapes as accurately as possible. These methods are usually very fast but cannot be used unless a rough initial alignment is available.

2.1. Related Work

Rigid problems aside, other types of registration exist. One of the most challenging problems today is that of nonrigid registration. In this case, the constraints based on Euclidean distance preservation are not valid. The extrinsic methods commonly used in rigid registration are unable to solve nonrigid problems. Instead of geometrical properties that are sensible to nonrigid deformations, these methods often use topological information and the intrinsic properties of the objects to align the shapes properly. However, some of these methods also display good performance in rigid registration. We review some of them in this article.

Some examples of the state of the art in nonrigid registration are heat kernel signature (HKS) [Sun et al. 2009], which computes the heat diffusion over a surface, and wave kernel signature (WKS) [Aubry et al. 2011], which calculates the average probability of measuring a quantum mechanical particle at a specific location. Both methods can be used in other intrinsic searching strategies, shown in Ovsjanikov et al. [2012], where the authors presented an approach called *Functional Maps*. This approach uses a generalization of the notion of map and looks for correspondences between real-valued functions rather than points on the shapes. Other examples are a nonrigid improvement of the 3D Shape Context descriptor presented by Kokkinos et al. [2012] and a nonrigid application of curve skeletons Zheng et al. [2010].

Due to the width of the registration research field, it is necessary to narrow the scope of this review. For this reason, we focus on rigid registration alone.

3. PROBLEM DEFINITION

The variety of registration applications produces divergent notations throughout the literature. Similar notions often receive different names, and in some cases, formal definitions of commonly used concepts are not widely available. To improve the readability of this review and unify related concepts in this section, we introduce a formal definition of the problem.

3.1. Input Data

Each of the many applications that use registration techniques has its preferred data type. Essentially, there are three types of input data used in the literature: point clouds, triangular meshes, and volumetric data. The simplest is the former, which is a collection of 3D points with no other information. The second is composed of a point cloud and connectivity information between points, usually presented as a graph. The most commonly used format is a triangular mesh (i.e., a Delaunay mesh). Volumetric data is often used in medical imaging (MRI, tomography, etc.) due to the nature of acquisition. These types of data are considered to be easily processed in parallel. In this article, we focus on point clouds and meshes.

The input data of the registration problem thus consists of two point clouds $\mathcal{A}$ and $\mathcal{B}$, being $\mathcal{A} = \{a_1, \ldots, a_n\}$ with $a_i = (x_{a_i}, y_{a_i}, z_{a_i})$ and $\mathcal{B} = \{b_1, \ldots, b_m\}$ with $b_i = (x_{b_i}, y_{b_i}, z_{b_i})$. Note how, in some cases, such as the reconstruction of an object from several views, more than two objects might be involved in the registration problem. As these problems can be reduced to a series of pairwise registration instances, we do not explicitly include these problems in this section. Whenever the problem requires the use of meshes, we name them $M_{\mathcal{A}}$ and $M_{\mathcal{B}}$. These meshes are graphs $M_{\mathcal{A}} = (\mathcal{A}, E_{M_{\mathcal{A}}})$, $M_{\mathcal{B}} = (\mathcal{B}, E_{M_{\mathcal{B}}})$, where $E_{M_{\mathcal{A}}}, E_{M_{\mathcal{B}}}$ (graph edges) contains relationship information between the points of the object (graph vertices).

When it comes to the use of meshes, nearly all functions aiming at describing shape in the vicinity of a point a_i are based on the *neighbors* of a_i. These neighbors might be defined in terms of Euclidean distance, requiring range-searching data structures for their computation, or in terms of a mesh, where the neighbors of a_i correspond to its adjacent points, connected by $E_{M_{\mathcal{A}}}$. Note how either notions do not always coincide, especially as meshes might be constructed using a variety of criteria. Nevertheless, meshes are used often, as they provide fast and convenient access to neighbors.

3.2. Desired Output

The registration problem aims at finding a rigid transformation[9] $\mu : \mathbb{R}^3 \longrightarrow \mathbb{R}^3$ that brings set $\mathcal{A}$ as close as possible to set $\mathcal{B}$ in terms of a designated set distance. A commonly used distance is the root mean squared distance (RMSD) defined as follows:

$$\mathrm{RMSD}(X, Y) = \sqrt{\frac{\sum_i^n |x_i - y_i|^2}{n}}. \tag{1}$$

In applications where only partial matches are expected, it is sometimes desirable to fix an upper threshold *thr* for the distance between a_i and $\mu(b_j)$ so that points without correspondences do not influence the measure. The matching process returns a set of correspondences C between $\mathcal{A}$ and $\mathcal{B}$ where correspondences with distance farther than *thr* are discarded:

$$\begin{gathered} C = \{(a_i, b_j) \text{ with } a_i \in \mathcal{A}, b_j \in \mathcal{B} \\ \text{holding } \forall b_k \in \mathcal{B}\ d(a_i, b_k) \geq d(a_i, b_j) \text{ and } d(a_i, b_j) < thr\}. \end{gathered} \tag{2}$$

Then we redefine the RMSD as

$$\mathrm{RMSD}(\mathcal{A}, \mu(\mathcal{B})) : a_i, b_j \in C = \sqrt{\frac{\sum_C d(a_i, \mu(b_j))^2}{|C|}}, \tag{3}$$

where $\mu(b_j)$ is the nearest neighbor point of a_i and $|C|$ is the cardinality of set C.

3.3. Detectors and Descriptors

Two very important steps of the registration problem are *detection* and *description*. In the first, the goal is to select those points of the sets that are more distinctive according to a chosen criterion (in most cases, the shape of the object). Besides, descriptors aim at encoding the shape around a point in terms of a set of numerical values. Consequently, although detectors and descriptors are focused on different targets, both are based on the same key issue: the local shape of the input data. The key points detected from an input data are selected according to the salience and uniqueness of the descriptor value at these points. In this review, we present both steps in separate sections (Sections 4 and 5), as most papers focus only on the one aspect. In this section, we highlight that both topics are very close, because both are based on the study of the shape around a certain point.

We define this *shape function* of a certain point a_i as $f^D(a_i) : N_{a_i} \subset \mathbb{R}^3 \longrightarrow P(\mathbb{R})$ being N_{a_i} the neighborhood of a_i, where the superscript D identifies the method. $P(\mathbb{R})$ is the power set of $\mathbb{R}$ (e.g., the set of all subsets of $\mathbb{R}$). For each point a_k in N_{a_i}, $f^D(a_i)$ outputs a set of real values corresponding to the shape of N_{a_i} around a_i. Usually, the same descriptor function is used for the two sets involved in the matching $\mathcal{A}$ and $\mathcal{B}$. To avoid some cumbersome notation, from now on we will obviate this particular set N_{a_i}

[9] Holding $d(a_i, b_j) = d(\mu(a_i), \mu(b_j)) \forall a_i, b_j \in \mathbb{R}^3$, $d()$ being the Euclidean distance.

and refer to these functions as $f^D(a_i) : \mathbb{R}^3 \longrightarrow P(\mathbb{R})$. Some examples of these functions are as follows:

—$f^{\text{PS}}(a_i) : \mathbb{R}^3 \longrightarrow \mathbb{R}^2 \times \cdots \times \mathbb{R}^2$ for the Point Signature descriptor [Chua and Jarvis 1997], where $f^{\text{PS}}(a_i)$ is a list of paired values for each point a_i in $\mathcal{A}$.
—$f^{\text{SI}}(a_i) : \mathbb{R}^3 \longrightarrow$ *2D histogram* for the Spin Image descriptor [Johnson 1997], where $f^{\text{SI}}(a_i)$ is a distribution histogram of the points in the neighborhood of a_i.
—$f^{\text{HKS}}(a_i) : \mathbb{R}^3 \longrightarrow \mathbb{R}$ for the HKS descriptor [Sun et al. 2009], where $f^{\text{HKS}}(a_i)$ is the value of the heat diffusion function around a_i.

In the detection step, the most distinctive points are selected according to f^D. A threshold is often used for this task, and only the points that satisfy this threshold are kept. We define this subset of selected points as $S_{\mathcal{A}} \subset \mathcal{A}$ that will be used in the rest of the pipeline. In the description step, we use f^D to obtain a value that represents the shape around the point. We define a Boolean correspondence function $cf^D : S_{\mathcal{A}} \times S_{\mathcal{B}} \longrightarrow boolean\{0, 1\}$ that checks whether or not the descriptor values at two given points are close enough for the shapes around the points to be considered the same:

$$cf^D(a_i, b_j) = \begin{cases} \text{TRUE} & \text{if } f^D(a_i) \approx f^D(b_j) \\ \text{FALSE} & \text{if } f^D(a_i) \neq f^D(b_j). \end{cases} \tag{4}$$

As an example, we present the correspondence function cf of the Spin Image descriptor:

$$cf^{SI}(a_i, b_j) = \begin{cases} \text{TRUE} & \text{if } ||\alpha_{a_i} - \alpha_{b_j}|| < \epsilon \quad \text{and} \quad ||\beta_{a_i} - \beta_{b_j}|| < \epsilon, \\ \text{FALSE} & \text{otherwise} \end{cases} \tag{5}$$

where $f^{SI}(a_i) = (\alpha_{a_i}, \beta_{a_i})$ and $f^{SI}(b_j) = (\alpha_{b_j}, \beta_{b_j})$.

3.4. The Computation of Output Motions

Throughout the literature, all motions μ considered as candidate outputs for the registration problem are computed using a number of point correspondences. First, points in sets $\mathcal{A}$ and $\mathcal{B}$ (a_i, b_j) are identified as possibly corresponding points. If a descriptor is being used, (a_i, b_j) must hold $cf^D(a_i, b_j) = \text{TRUE}$. Then, once a number of these "corresponding couples" have been identified, a motion is computed following specific criteria. Usually, the criterion used is the least squares distances between sets. In a typical scenario [Díez et al. 2012], three corresponding couples, $(a_{i_1}, b_{j_1}), (a_{i_2}, b_{j_2}), (a_{i_3}, b_{j_3})$, are identified, and μ is usually the rigid transformation holding that the RMSD between sets $\{a_{i_1}, a_{i_2}, a_{i_3}\}$ and $\{\mu(b_{j_1}), \mu(b_{j_2}), \mu(b_{j_3})\}$ is the minimum possible. To determine a 3D rigid transformation, at least three point correspondences are mandatory (although more might be used [Winkelbach et al. 2006; Aiger et al. 2008]). The number of point correspondences varies for other types of motions (e.g., only one point correspondence is needed to determine a 3D translation).

These sets of points used to compute candidate motions are a commonly used concept. Henceforth, we will refer to this concept as a base.

A *base* $B_{\mathcal{A}} = \{a_{i_1}, \ldots, a_{i_k}\} \subset S_{\mathcal{A}}$ of set $\mathcal{A}$ stands for the set of k points used to determine a rigid transformation that is a candidate to be the output of the registration problem. Each point of $B_{\mathcal{A}}$ must have a correspondence point in an analogous $B_{\mathcal{B}} = \{b_{j_1}, \ldots, b_{j_k}\} \subset S_{\mathcal{B}}$ holding $cf^D(a_{i_l}, b_{i_l}) = \text{TRUE}$.

Note how although only a few points from each set are usually considered when computing candidate motions, the measure of the proximity of the two sets is computed using all points in the sets or, at least, all matched points.

3.5. Problem Statement

A formal summary of this section follows:

—Given two point sets $\mathcal{A} = \{a_1, \ldots, a_n\}$ and $\mathcal{B} = \{b_1, \ldots, b_m\}$ and a shape function $f^D : \mathbb{R}^3 \longrightarrow P(\mathbb{R})$ with correspondence function $cf^D : S_\mathcal{A} \times S_\mathcal{B} \longrightarrow boolean\{0, 1\}$.
—Let $S_\mathcal{A} \subset \mathcal{A}$ and $S_\mathcal{B} \subset \mathcal{B}$ be the points that are most distinctive in term of a shape descriptor function $f^D : \mathbb{R}^3 \longrightarrow P(\mathbb{R})$.
—A solution to the rigid registration problem is a rigid transformation μ holding:
  —RMSD($\mathcal{A}$, $\mu(\mathcal{B})$) is minimum.
  —There exist two bases $B_\mathcal{A} = \{a_{i_1}, \ldots, a_{i_k}\} \subset S_\mathcal{A}$, $B_\mathcal{B} = \{b_{j_1}, \ldots, b_{j_k}\} \subset S_\mathcal{B}$ holding that for all corresponding couples, $cf^D(a_{i_l}, b_{j_l}) = \text{TRUE}$.

4. DETECTORS

To reduce both the computation time and the number of points to be considered, the most common strategy is to use only those points that can effectively contribute to finding a good enough solution. In other words, the goal is to obtain a subset of points that maintain the object shape characteristics as far as possible. This is a rapidly growing research field, motivated by 3D shape retrieval problems [Shilane et al. 2004; Tangelder and Veltkamp 2004; Iyer et al. 2005; Bustos et al. 2005; Lian et al. 2012]. This step is also called *filtering*. Several criteria exist to decide which points should be kept and which points should be discarded. Note that many methods explained here are used in combination with a descriptor, usually presented under the same name. In this section, we introduce the most remarkable methods in the literature according to the results presented in each paper and in different reviews and benchmarks [Bronstein et al. 2010; Boyer et al. 2011; Salti et al. 2011; Dutagaci et al. 2012; Yu et al. 2013].

4.1. Normal Space Sampling

Rusinkiewicz and Levoy [2001] reviewed several methods, such as uniform [Turk and Levoy 1994] or random [Masuda et al. 1996] sampling. The main problem with these methods is that the selection of points does not depend on surface characteristics. Dealing with smooth models with small irregularities (e.g., a plane), the process might result in sampling many points that essentially contain the same information in terms of normal vectors. For this reason, the authors introduced the *Normal Space Sampling* (NSS) method. This strategy consists of (1) grouping points in "buckets" according to the angles between their normal vectors (considered in the unit sphere) and the coordinate axes, and (2) sampling uniformly over the resulting buckets, providing a downsampling of the points with more "frequent" normal vectors.

Díez et al. [2012] presented an improvement of NSS called *Hierarchical Normal Space Sampling* (HNSS). This method groups points hierarchically, according to the distribution of their normal vectors with each level in the hierarchy representing a NSS instance. The search for correspondences then proceeds hierarchically between points of the two sets. The huge search space is navigated taking advantage of geometric information until a solution is found. The authors use a RANSAC-based method inside the hierarchical structure to find a transformation that roughly aligns the two point sets. A significant reduction in computation time is observed when compared to pure RANSAC-based methods.

4.2. Maximally Stable Volumes

The concept of Maximally Stable Volumes (MSVs) [Donoser and Bischof 2006] is a 3D extension of Maximally Stable Extremal Regions (MSER) [Matas et al. 2004]. MSV detects the most stable regions in a volume across different binary thresholds. Given a volumetric shape, the points inside the regions that remain visible under a set of binary thresholdings will share good key points for a registration process.

A 3D volume can be interpreted as a weighted graph. Each voxel of the volume represents one node in the graph, and its value (e.g., the intensity value of MRI data) is the weight of this node. The connectivity between nodes is given by the spatial neighborhood of the voxels. A level set L_w of a weighted graph contains the set of nodes with a weight above a given threshold w. The connected nodes within the same level are grouped in connected components. To find an MSV, the authors propose the use of a rooted data structure, namely a *component tree*. A component tree of a weighted graph is an ordered representation of the graph. The component tree of a 3D volume has connected volumes C_i^w as a tree nodes. Each level of the component tree contains the connected volumes of a specific level set L_w at weight w. The MSVs are identified as the connected volumes with the highest stability along a thresholding process thorough all levels of the component tree. There are different algorithms for computing the component tree, but the most efficient is the algorithm proposed by Najman and Couprie [2004] that runs in quasilinear time.

There are different options for the key-point selection, such as a random sampling of the surface of the MSV or a selection of the center of the ellipsoid contained inside the MSV as a key point.

In Yu et al. [2013], MSV is tested against other detectors like Harris 3D, SURF, or MeshDoG, obtaining the best performance results and being robust to noise and rotation. Compared to the other methods, MSV detects few key points in the input surface, but the ratio of correspondences between two registered shapes is at least twice as big as the other detectors. The main drawback is the computation time, as the search algorithm for stable regions is less efficient in 3D than in 2D.

4.3. Heat Kernel-based Features

As we will see in Section 5, HKS is a point descriptor presented by Sun et al. [2009]. However, the authors use the same concept in the mentioned reference: the heat diffusion in a surface over a temporal domain as a key-point detector.

The nature of this method makes it possible to use the HKS as a shape function f^{HKS} to detect the parts of the shape with zones that are more salient in terms of descriptors, like zones with high curvature. High values of f^{HKS} identify the key points of a shape.

HKS is one of the best detectors in the literature due to its high repeatability results (≈90%). HKS tends to return fewer key points than others detectors but with high distinctiveness. In the SHREC 2010 benchmark [Bronstein et al. 2010], HKS obtains the best results, and in Dutagaci et al. [2012], when it is compared with human-generated ground truth, it performs much better than the other methods—close, in fact, to human selected key points. Additionally, it can also be used for nonrigid registration.

4.4. MeshDoG

Zaharescu et al. [2009] presented a point detector based on the difference of Gaussians (DoG) operator (Figure 2). MeshDoG finds the extrema of the Laplacian of a scale-space representation of any scalar function defined on a discrete manifold.

Assuming a uniformly sampled triangulated mesh $M_{\mathcal{A}}$ as input data, the authors find the key points using the DoG operator. For each point $a_i \in M_{\mathcal{A}}$, the extrema of the Laplacian function are found across scales using a one-ring neighborhood N_{a_i}. Then, the feature points are selected as the maxima of the scale space across scales. Finally, only 5% of feature points are selected to make this detection step more accurate. Only those feature points exhibiting corner characteristics are considered.

MeshDoG achieves high repeatability results (≈85%) [Boyer et al. 2011], being robust to rigid transformation and scale modifications.

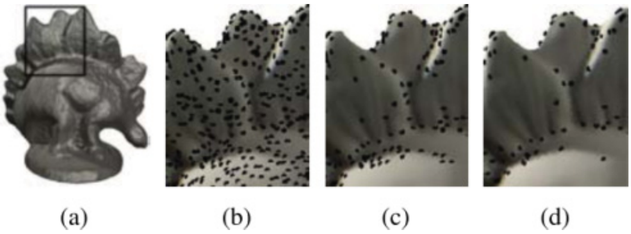

Fig. 2. Three steps of MeshDoG algorithm applied on a mesh (a), scale-space extrema detection (b), thresholding (c), and corner detection (d). Image taken from Zaharescu et al. [2009].

4.5. Intrinsic Shape Signatures

Intrinsic Shape Signatures (ISS) [Zhong 2009] is a point descriptor that possesses its own detection method. As we will see in Section 5, ISS uses the eigendecomposition of the neighborhood's covariance matrix of a point to describe it. These values are used to select the most representative points $S_{\mathcal{A}}$ and $S_{\mathcal{B}}$ from the point clouds. Key points are selected as the points with large 3D variations in their neighborhood. These variations are measured using the smallest eigenvalue of the covariance matrix of its spherical neighborhood.

ISS demonstrates high repeatablity results ($\approx$70%) with processed data, even in noisy scenes, identifying few but strong key points. However, with real data, these results drops to approximately 30%.

Another detector, Key-Point Quality (KPQ) [Mian et al. 2010], is quite similar to ISS. KPQ also uses the neighborhood's eigendecomposition of the covariance matrix to establish a local reference frame (LRF). As a major difference from ISS, KPQ defines a Key-Point Quality measure of each point based on the principal curvatures of the local surface within a neighborhood. A smoothed surface S is fitted over the original data. S is divided in a grid to sample the surface. Principal curvatures k_1 and k_2 and the Gaussian curvature $K = k_1 k_2$ of each sample of S are used to calculate the key-point quality. The method selects the key points with high quality value and, with processed data, this method performs worse than ISS. It obtains better results, however, with real models.

4.6. Harris 3D

Sipiran and Bustos [2011] presented a 3D version of the Harris operator. The idea is to apply the Harris operator in a 2D projection of the points without losing relevant information. To find the best projection, where the points exhibit a good spread, the authors translate and rotate the set of points of $\mathcal{A}$ according to the following criterion: for each point $a_i \in \mathcal{A}$, a neighborhood N_{a_i} is defined. The centroid of N_{a_i} is computed and all points in $\mathcal{A}$ are translated so that the centroid coincides with the origin of the coordinate system. Then, a fitting plane to the translated points is computed. Authors apply principal component analysis (PCA) to the set of points and choose the eigenvector with the lowest associated eigenvalue as the normal of the fitting plane. Afterward, they rotate the set of points until the normal of the plane coincides with the z-axis. Finally, the resulting xy plane (2D projection) is used to calculate the derivatives. These derivatives are computed using a six-term quadratic surface (paraboloid) fitted to the set of transformed points. The Harris operator value in the studied point is calculated with

$$f^H(a_i) = \det(E) - k(\mathrm{tr}(E))^2, \tag{6}$$

where E is a matrix calculated from the points using the quadratic surface mentioned earlier. The vertices with highest Harris values are considered feature points, obtaining a constant number of vertices.

Due to the simplicity of the algorithm, Harris is faster. However, it is not robust to noise because the corner detector methods are sensible to the perturbations of the surface [Bronstein et al. 2010]. Although the method detects many key points, the ratio of correspondences is small, around 20%, and decreases considerably when the noise increases.

5. DESCRIPTORS

The shape function f^D, also frequently referred to simply as the descriptor of a_i, can be defined as a set of values representing the shape characteristics of object $\mathcal{A}$ around a_i. A desirable characteristic for 3D rigid registration is for this representation to be invariant under translation, rotation, and scaling.

In terms of the number of papers published, descriptors are the most active research field in the registration pipeline. We classify many existing approaches according to certain common characteristics. Following the work of Tombari et al. [2010], we arrange the approaches in *signatures* and *histograms*. The former include methods that offer a numerical result as a descriptor of a given point. The latter compute a histogram.

Another distinguishing factor is the type of the input data. Most methods work with point clouds without any added structure ($\mathcal{A}$), but some methods need to produce richer representations, like meshes ($M_{\mathcal{A}}$). A triangulation with good shape properties, such as the Delaunay triangulation, where the distribution between vertexes, edges, and faces is approximately regular, comes with a high computational cost ($O(n^2)$). This cost, however, is incurred only once in a preprocessing step.

Another factor that we use for the discussion in this section are reference frames. In papers like Zhong [2009] and Tombari et al. [2010], the authors note the importance of achieving a good LRF to improve the accuracy of the detectors/descriptors. This accuracy stems mainly from having an unambiguous LRF for every point, allowing for detailed descriptions of local shapes. As we will see in Section 8, one possible drawback of this approach lies in its sensibility to noise, especially occlusions. These factors greatly perturb the local neighborhoods of points and thus affect the computations of LRF.

Finally, we classify the methods according to their geometrical or topological nature. Although topological methods are primarily used in nonrigid registration, they are also used in rigid registration.

To make this classification easier, we present the methods of this section with the following acronyms:

—*S/H*: Signature-based or histogram-based method
—*P/M*: Point cloud or mesh as an input data type
—*RF/nRF*: A reference frame is used or not
—*G/T*: Geometrical or topological method.

As mentioned previously, even though nonrigid registration methods are out of the scope of this review, we include some of them because they obtain successful results when on rigid problems.

5.1. Principal Curvature [S,P,nRF,T]

Principal Curvature stands for the maximum and the minimum curvature of the surface at a given point. Feldmar and Ayache [1996] proposed using it as a descriptor. In this approach, key points are described by the principal curvatures (k_1, k_2), the normal vector of the point ($\vec{n}$), and the principal directions ($\vec{e_1}, \vec{e_2}$) corresponding to the

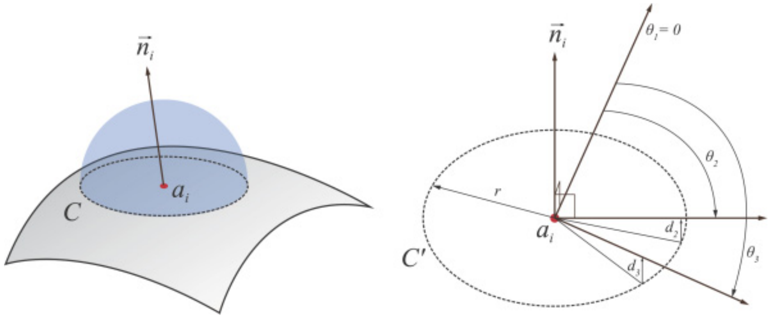

Fig. 3. Representation of a Point Signature. Left: Intersection between the surface $\mathcal{A}$ and the sphere centered at a_i, giving a curve C. Right: Distances and angles of different points in C projected to C'.

principal curvatures. To search for correspondences between two point clouds, the algorithm considers an initial point a_i on surface $\mathcal{A}$ with a descriptor $f_{a_i}^{PC} = (a_i, \vec{e_{1i}}, \vec{e_{2i}}, \vec{n_i})$, and a set of possible candidates on the second surface $\mathcal{B}$, each one with a descriptor $f_{b_j}^{PC} = (b_j, \vec{e_{1j}}, \vec{e_{2j}}, \vec{n_j})$, similar to $f_{a_i}^{PC}$. Two rigid transformations are defined, D and D', where D aligns $f_{a_i}^{PC}$ with $f_{b_j}^{PC}$ and D' aligns $f_{a_i}^{PC}$ with $f_{b_j}^{PC'} = (b_j, -\vec{e_{1j}}, -\vec{e_{2j}}, \vec{n_j})$. Note that both rigid transformations, D and D', are computed because there is no way to choose between them, as the direction of $\vec{n_i}$ is ambiguous. Afterward, the transformation matrix that aligns both views is computed and evaluated. The authors consider all correspondences between $\mathcal{A}$ and $\mathcal{B}$ at a distance smaller than a certain threshold. If not enough correspondences are found, the algorithm chooses another initial point a_i and iterates. Otherwise, the alignment is computed using the available correspondences. This method is also used in nonrigid registration because surface curvatures are invariant to isometric deformations [Feldmar and Ayache 1996].

The main problem of this algorithm is that only one correspondence is used to compute the rigid transformation. Considering that the algorithm stops when it finds a good correspondence, other possible correspondences might not be considered. Better alignments might be missed if the motion found is affected by noise or occlusion [Salvi et al. 2007].

5.2. Point Signature [S,P,RF,G]

Point Signature is a descriptor introduced by Chua and Jarvis [1997]. For a point a_i on a surface $\mathcal{A}$, a sphere of radius r centered at a_i is considered. The intersection between the surface $\mathcal{A}$ and the sphere determines a curve C. This curve is projected on a plane tangent at a_i and perpendicular to $\vec{n_i}$, giving a contour C'. Then, taking a_i as center of coordinates, the authors define an orientation axis with the normal vector $\vec{n_i}$, a reference vector $\vec{n_1}$, and the cross product between them. Each point in C will be described by a signed distance between itself and its projection on C' and the rotation angle from the reference vector n_1. The Point Signature of a_i will be expressed as the set of distances and angles of the points on C. To find correspondences between two point clouds, Point Signatures of all points are compared following the example seen in function 5. Figure 3 shows a representation of the descriptor.

Although the matching process is fast, the cost of the intersection between the surface and the sphere requires the use of range-searching data structures.

5.3. Spin Image [H,M,RF,G]

In 1997, Johnson presented a descriptor based on the position of the neighbors of a given point [Johnson 1997; Johnson and Hebert 1999]. The authors consider a point a_i and its

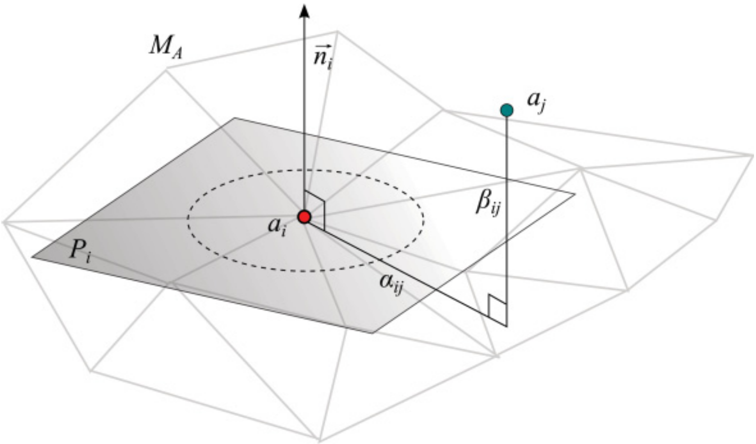

Fig. 4. Spin Image representation: an oriented point basis created at a vertex in a surface mesh. The position of the oriented point is the 3D position of the vertex, and the direction of the oriented point is the surface normal at the vertex.

associated normal vector $\vec{n_i}$. They define a plane P tangent to a_i and perpendicular to $\vec{n_i}$. The neighborhood N_{a_i} around a_i will be registered based on two variables: distance α between each point and the normal vector $\vec{n_i}$, and the distance β between each point and the tangent plane P (Figure 4). The following functions are used to calculate both parameters:

$$\alpha = \sqrt{||x - a_i||^2 - (\vec{n_i}(x - a_i))^2}, \tag{7}$$

$$\beta = \vec{n_i}(x - a_i). \tag{8}$$

A table called *Spin Map* is generated with this information, where each point x around a_i is projected according to α on the x-axis and β on the y-axis. Each cell of the Spin Map contains the number of points belonging to the corresponding region. The generation of the shape function f^{SP} can be visualized as a rotating sweep over $\vec{n_i}$, where all Spin Maps are accumulated. Then, to find the correspondences between two different shapes, Spin Images are compared counting the points falling in the corresponding bins of both Spin Images.

This method is invariant to rigid transformations. It is, however, sensitive to symmetries and noise. Another problem is that the result of the method depends largely on the resolution used. Carmichael et al. [1999] proposed an improvement called *Face-based Spin Image* to solve these problems, where the numbers of points in each Spin Image are uniformly assigned.

Spin Image is the base of numerous recent approaches. Two examples are ISS [Zhong 2009] and SHOT [Tombari et al. 2010]. Both methods stress the importance of choosing a good reference frame. These reference frames are chosen via eigendecomposition of the covariance matrix from neighboring points. The eigenvectors with higher eigenvalues are used as the axes of the reference frame. Then, the authors use this reference frame to compute a version of Spin Image. ISS makes an occupational histogram of points inside the supporting sphere neighborhood around the point. SHOT makes a histogram of differences between the point and the neighbors inside the supporting sphere. Although both methods obtain satisfactory results with processed data, as we see in Salti et al. [2011], neither achieves sound results with real data.

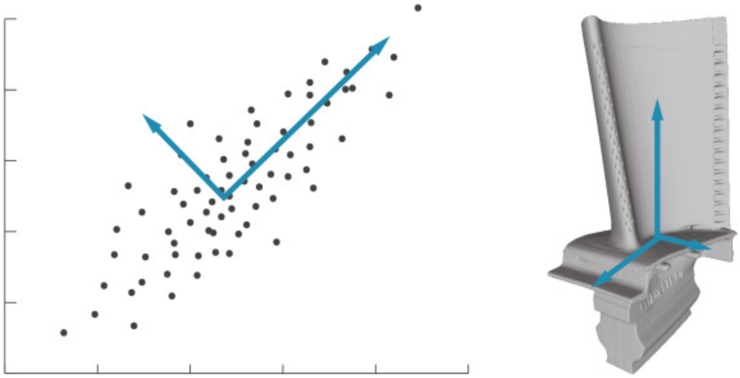

Fig. 5. Examples of PCA in 2D (left) and 3D (right) point clouds.

Zhang et al. [2012] presented *Improved Spin Image* (ISI) using angle information between the normals of feature points and neighboring points. The β parameter is replaced by signed angles. This method can be explained as a distribution of the angles among different rings. The authors claim that their descriptor outperforms both the classic implementation of Spin Image and also the SHOT method.

Besides these approaches, other works following the path opened by Spin Image are *Spherical Spin Image* [Ruiz-Correa et al. 2001], *Local Surface Patches* [Chen and Bhanu 2007], and *Scale-Invariant Spin Image* [Darom and Keller 2012].

5.4. Principal Component Analysis [S,P,RF,G]

Initially, the theoretical basis of this method was presented by Pearson [1901] to transform a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called *principal components*. It was exported to other fields, such as statistics, computer vision, and computational geometry. In point cloud registration, this method is used to find the principal axes that describe the shape of a point cloud (Figure 5). Given two point clouds $\mathcal{A}$ and $\mathcal{B}$ from the same object, if the main axes are coincident, we can find a transformation that aligns both coordinate systems.

Chung et al. [1998] presented a registration algorithm based on PCA, using the covariance matrix to determine the transformation μ between two point clouds. This method can also be considered a searching strategy, due to the global understanding of the algorithm, because it finds the principal component of all points in the point cloud. However, there are many other algorithms that implement local PCA and obtain principal components of local neighborhoods, considering it a point descriptor.

There are many approaches that use PCA. Pottmann et al. [2009] and Yang et al. [2006] use PCA to find principal components of a local sphere neighborhood. Sipiran and Bustos [2011] and Darom and Keller [2012] use PCA as an interest point detector. Johnson [1997] uses PCA to compress Spin Images, whereas Körtgen et al. [2003] use it to find the orientation of 3D shapes. Liu and Ramani [2009] presented an improvement of PCA for rigid and nonrigid registration, robust to noise and outliers, using the least median of squares (LMS) technique.

PCA is a very fast method, but it has some drawbacks that constrain its use in some practical applications. The algorithm needs large set overlap ($\geq 50\%$) to find good correspondences and symmetries in the surface. Furthermore, the presence of noise in the original point cloud may influence the alignment [Bailey 2012].

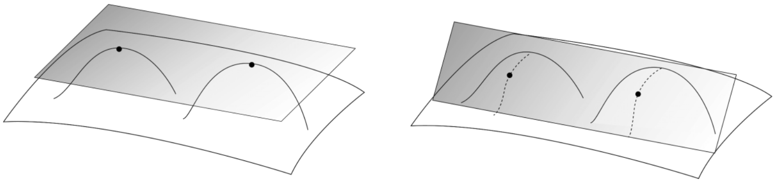

Fig. 6. Representation of a bitangent plane from Line-based method, rolling over a surface and describing two bitangent curves. This picture is taken from Vanden Wyngaerd and Van Gool [2002].

5.5. Line-based algorithm [S,P,nRF,G]

Vanden Wyngaerd and Van Gool [2002] presented a descriptor based on bitangent curves. The reason for using these types of curves instead of typical surface curvatures is that they are easier to calculate using dual space. Using two bitangent points lying in the same plane and rolling the plain over the surface, we obtain two bitangent curves that are used as a shape descriptor f^{LB} (Figure 6).

The key to this method is that in transforming the range images on dual space, the bitangent points of the surface are coincident. This transformation decreases the computing time and improves the robustness. The main problem of this method is that in some cases, the number of bitangent curves may be insufficient for the achievement of a good registration result. Moreover, noise hampers the search for correspondences.

5.6. 3D Shape Contexts [H,P,nRF,G]

Körtgen et al. [2003] presented a 3D extension of *2D Shape Contexts*. This method consists of describing a certain point in relation to the other points in the object, and not only the points in the neighborhood around it. Due to the size of the datasets, this algorithm only uses random sampled points instead of the full-sized data. The complexity of this detection step is $O(Slog(n))$, with S being the number of samples taken from n points.

Given a sampled point a_i, the method finds the vectors from a_i to all other sampled points (Figure 7). These vectors express the appearance of the entire shape in relation to the reference point a_i. For each point on the sampled set, the shape function f^{3DSC} is defined as a coarse histogram of the relative coordinates of the remaining $N-1$. To create this histogram, a sphere-space division is used, with center at a_i. This sphere is divided into bins. This space discretization is enough to obtain a robust descriptor.

The last part looks for correspondences. This is the most computationally expensive part of the algorithm. Two specific techniques are used to match the descriptors between two different shapes: *Local Matching* and *Global Matching*. The first one combines three concepts: shape, appearance and position. The goal is to obtain an invariant shape descriptor for each sampled point. For Global Matching, two different strategies are used: hard and soft assignments. Hard assignments stand for a one-to-one searching method and require high computational cost. However, this cost is reduced on soft assignments by carrying out a preselection of the candidates in second shape for each point in the first shape. Using this approach, the authors achieve a cost of $O(n_1 n_2)$ instead of $O(n^3)$ from hard assignments (n_1 and n_2 are the number of samples for each shape).

The authors claim that the method is robust to noise, topological, and geometrical artifacts and invariant under transformations. These claims are backed by an experimental study carried out using processed data. One question that remains is whether this method may have some problems with real scanned data, especially regarding

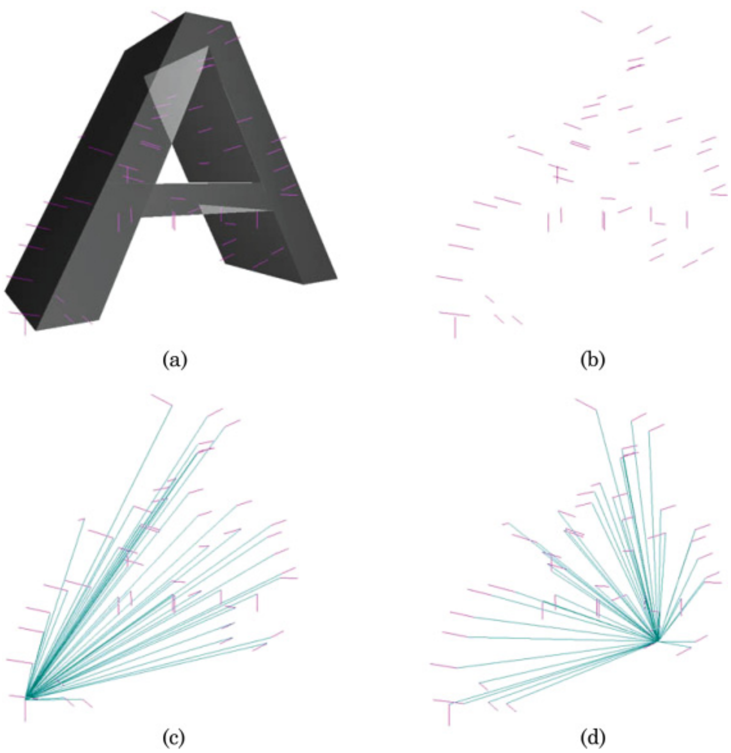

Fig. 7. Example of 3D Shape Contexts taken from [Körtgen et al. 2003]. (a) Mesh with 50 samples. (b) Just the 50 samples. (c) 49 vectors originating from one sample point. (d) 49 Vectors originating from another sample point.

shapes with a low ratio of overlapping. Questions naturally arise from the fact that the descriptor is computed using sampled points from the entire model.

5.7. Dynamical Systems [S,P,nRF,T]

Dey et al. [2003] presented a topological shape segmentation method called *dynamical systems*. Instead of focusing on local geometric properties of the shape, this approach identifies and segments the main sections of the shape from a global point of view. A single point represents an entire segment, where the weight of this point is the volume of the segment. First, given a set of points $\mathcal{A}$, the Voronoi diagram and the Delaunay triangulation are computed. Following the theory of the flow induced by a shape, critical points are selected. These points are defined to be the intersection points of the Voronoi objects with the Delaunay objects. For each critical point a_i, a stable manifold $S(a_i)$ is defined as the set of points that flow into a_i, grouping a set of Delaunay tetrahedra. Thus, the closure of these stable manifolds stands for the features of the shape. Then, the authors defined a *signature* as a set of features of $\mathcal{A}$. Each feature yields a representative point r, which is the weighted average of the centroids of all Delaunay tetrahedra from each feature. The weight of r is the volume of the feature. Finally, the matching process is performed by computing the similarity of the signatures of two shapes.

Few signatures are used to align two shapes. However, the cost of the algorithm is $\Theta(mn)$, which is a drawback if more signatures are needed. Furthermore, the authors did not test the method with real or noisy data.

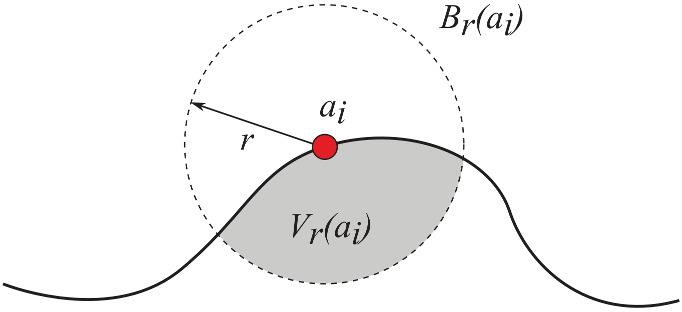

Fig. 8. 2D representation of the integral invariant descriptor.

5.8. Integral Invariants [S,M,RF,G]

The descriptors based on differential geometry, such as curvatures [Do Carmo 1976; Porteous 2001], are not robust to noise and perturbations, and require data smoothing and prior de-noising to achieve better results. This is problematic when working with real data, which usually contains noise and outliers. Integral Invariants produce a good solution for this problem, obtaining a descriptor based on the volume under the surface of an object. Given a point a_i on a surface $\mathcal{A}$, a sphere of radius r centered at a_i is computed. The method calculates the enclosed volume of the sphere $V_r(a_i)$ under the surface, and the value of this volume stands for the integral invariant descriptor f^{II}. Figure 8 shows a 2D representation of the process. To find corresponding points, the value of f^{II} is used for comparison purposes.

The first authors to study the applications of the volume descriptors were Manay et al. [2004]. The authors claim that the numerical differentiation methods applied on point descriptors are sensitive to noise. Integral invariant signatures, presented in his paper, are robust to noise, including discretization artifacts, and present a multiscale behavior. Pottmann et al. [2009] presented a stability analysis of integral invariants based on distance functions. This method is based on the PCA of local neighborhoods defined by kernel balls of various sizes. Yang et al. [2006] presented an experimental paper about integral invariants obtained by integration over local neighborhoods. This short paper compares the method based on applying PCA over a ball or sphere neighborhoods from Pottman with the *normal cycles* method [Cohen-Steiner and Morvan 2003] and the *osculating jet* method [Cazals and Pouget 2005]. The authors conclude that Integral Invariants are more robust to noise than the other methods while exhibiting the desired scaling behavior. However, the papers in question do not test the method with real data, without smoothing and de-noising. In this situation, with high amounts of perturbations, as well as with low overlapping regions and holes, integral invariants may not produce satisfactory results.

Pottman et al. [2009] proposed three different ways to compute integral invariants: the Fast Fourier Transform–based method, the octree-based method, and the triangulation-based method. All of these methods generate running times of the same order.

5.9. Curve Skeleton [S,P,nRF,T]

Curve-skeleton or skeletal graphs were introduced by Blum [1967]. This method consists in describing a shape by a thinned representation (Figure 9). A skeleton (stick figure) of the shape is extracted and converted to a skeletal graph that preserves the topological properties of the shape. Then, the graphs of the two objects being matched

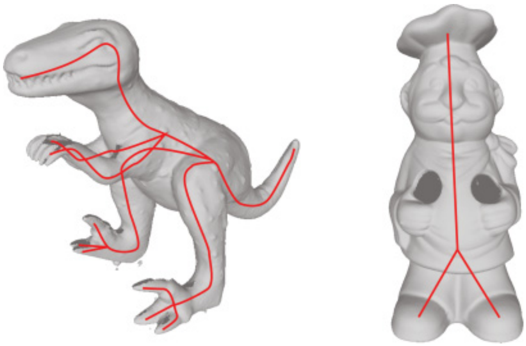

Fig. 9. Curve skeleton representation of T-Rex and Chef models.

are compared to register two different shapes. This method can be used either with point clouds, meshes, or volumetric data.

More recently, Cornea et al. [2007] presented a thorough state-of-the-art example of curve-skeleton approaches. According to Cornea et al. [2007], curve-skeleton methods have many useful properties for shape registration, such as homotopy, invariance under isometric transformation, robustness, and efficiency, among others. One of the most common ways to compute a curve skeleton is the Voronoi diagram, which represents the space subdivision of the shape. The internal edges and faces of the Voronoi diagram are used to approximate the skeleton. There are different matching methods that use the curve skeleton as a feature descriptor, such as Cornea et al. [2005], which uses a low-dimensional vector, whose components are based on the eigenvalues of the subgraph's (0,1) adjacency matrix, or Sundar et al. [2003], which consists of using a distribution-based similarity measure designed to evaluate dissimilarity between two multidimensional distributions.

A curve-skeleton descriptor can be used for matching incomplete point clouds [Tagliasacchi et al. 2009] and also for nonrigid registration [Iyer et al. 2005].

5.10. Point Feature Histograms [H,P,RF,G]

Point Feature Histograms (PFH) were presented by Rusu et al. [2008]. This method consists of extracting geometrical information from the neighborhood of a given point. Given a query point a_i from a point cloud $\mathcal{A}$, a sphere of radius r encloses the neighborhood N_{a_i}. All points lying inside the sphere are connected with the others via a fully interconnected mesh (Figure 10, left). For each point $a_j \in N_{a_i}$ with a normal vector $\vec{n_j}$, the algorithm selects another point $a_k \in N_{a_i}$ where the angle between $\vec{n_j}$ and the vector defined by $(a_k - a_j)$ is minimum. Basically, this means that the algorithm is focused on concave zones of the shape. For each pair of points a_j and a_k $(j \neq k)$, a reference frame called *Darboux uvn frame* is computed $(\vec{u} = \vec{n_j}, \vec{v} = (a_k - a_j) \times \vec{u}, \vec{w} = \vec{u} \times \vec{v})$. Then, the angular information is calculated with these functions:

$$\alpha = v \cdot \vec{n_k}, \tag{9}$$

$$\phi = \frac{u \cdot (a_k - a_j)}{||a_k - a_j||}, \tag{10}$$

$$\theta = arctan(w \cdot \vec{n_k}, u \cdot \vec{n_k}). \tag{11}$$

Finally, the algorithm builds a histogram divided into bins, where bin space is arranged covering all values of the features. For each query point a_i, a descriptor

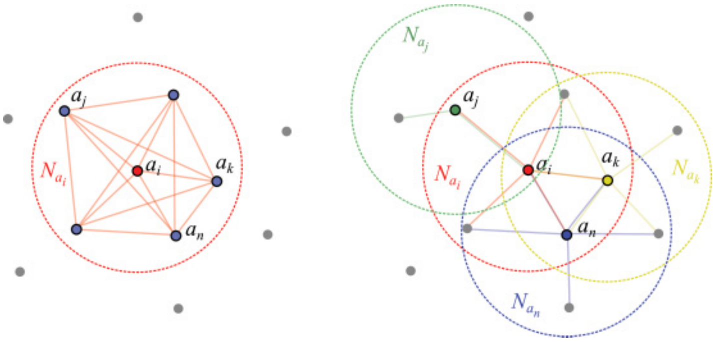

Fig. 10. Left: PFH representation. All relations between neighbors in N_{a_i} are taken into account to compute $f^{\text{PFH}}(a_i)$. Right: FPFH representation. Each point only uses its direct neighbors to compute his own SPFH. Then, the neighboring SPFHs are used to weight the final descriptor value of $f^{\text{FPFH}}(a_i)$. Note: Only three neighbors of a_i are shown in the figure for major clarity.

histogram is computed according to the value of angular information of each pair of neighbors (a_j, a_k).

The correspondences between points from different shapes are found by sampling a number of described points. For each sampled point in $S_{\mathcal{A}}$, a set of points in $S_{\mathcal{B}}$ are selected. From these, one point is randomly selected and a transformation μ computed. The quality of μ is evaluated by computing its error metric.

The main drawback of PFH is its high complexity $O(n \cdot m^2)$, where n is the number of points of $\mathcal{A}$ and m is the number of neighbors of each point. For this reason, the authors simplified the method and presented the Fast Point Feature Histogram (FPFH) [Rusu et al. 2009], which reduces the complexity to $O(n \cdot m)$. Given a query point a_i, instead of calculating the relationships between all points in N_{a_i}, only the direct neighbors of a_i are taken into account (see Figure 10). The angular information of these pairs of points are computed and a simplified point feature histogram (SPFH) is made using this information. Thus, each point in $\mathcal{A}$ has its own SPFH, computed only with its direct neighbors. Afterward, the SPFHs of the points inside the neighborhood N_{a_i} are used to weight the histogram of a_i, obtaining a $f^{\text{FPFH}}(a_i)$ (see Figure 10 right),

$$f^{\text{FPFH}}(a_i) = \text{SPFH}(a_i) + \frac{1}{m}\sum_{i=1}^{m}\frac{1}{w_m} \cdot \text{SPFH}(a_m), \tag{12}$$

where w_m is the distance between the query point a_i and its neighbor point a_m, used to weight the final value of FPFH(a_i).

FPFH is tested with real data with an overlap of approximately 45% and obtains good results combined with a specific searching strategy called *SAmple Consensus Initial Alignment* (SAC-IA) [Rusu et al. 2009].

5.11. MeshHOG [H,M,RF,G]

Zaharescu et al. [2009] dealt with local feature detection and description methods. The authors presented a 3D feature detector MeshDoG (difference of Gaussians) (see Section 4) and a 3D feature descriptor MeshHOG (histogram of oriented gradients) for uniformly triangulated meshes. The latter is a generalization of the histogram of oriented gradients (HOG) descriptor and uses two different parameters together to improve the surface registration: geometric and photometric information are extracted from the model to obtain more accurate results.

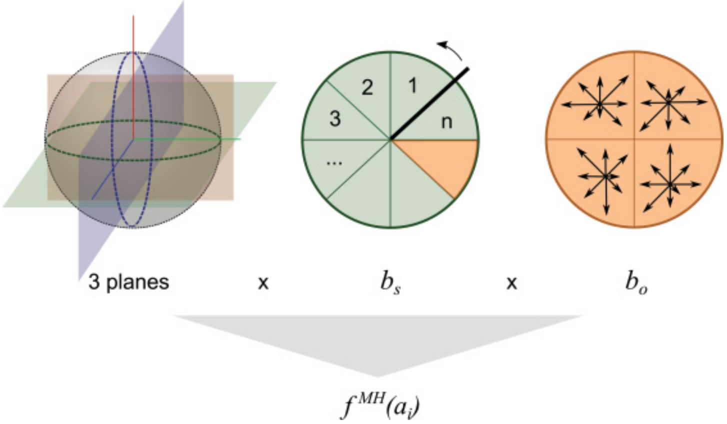

Fig. 11. Construction of HOG. Left: Choosing three orthogonal planes onto which to project the gradient vectors. Middle: Polar coordinate system used for creating histograms via binning of 2D vectors. Right: Example of a typical spatial and orientation histograms, using four spatial polar slices and eight orientation slices.

The shape function $f^{\mathrm{MH}}(a_i)$ of this method is computed using support regions, defined using a neighborhood ring N_{a_i}. To make the descriptor invariant to rotation, a local coordinate system is taken into account. For each vertex of the neighborhood of a_i, the gradient information is computed. These gradient vectors are projected onto the three orthogonal planes from the local coordinate system to make the representation of the descriptor more compact. For each of the planes, the authors compute a two-level histogram. First, the plane is divided into $b_s = 4$ polar slices. For each slice, the algorithm computes an orientation histogram, with $b_o = 8$ bins for each projected gradient vector. f^{MH} is finally computed by concatenating $b_s \times b_o$ for each of the three planes. We present the sequence of the algorithm in Figure 11.

The authors use an intuitive greedy heuristic algorithm as a correspondence function cf^{MH} for descriptor matching. Given two surfaces $\mathcal{A}$ and $\mathcal{B}$, two sets of descriptors $S_{\mathcal{A}} \subset \mathcal{A}$ and $S_{\mathcal{B}} \subset \mathcal{B}$ are extracted from both shapes. For each descriptor in $S_{\mathcal{A}}$, the algorithm finds the best correspondence in $S_{\mathcal{B}}$ in terms of Euclidean distance. Then, a cross validation is performed by checking, for each descriptor in $S_{\mathcal{B}}$, the best correspondence in $S_{\mathcal{A}}$. The total cost of the matching process is $O(n^2)$.

The running time of the algorithm depends on the size of N_{a_i}. The authors conclude that the descriptor is robust under rigid transformations and outperforms the traditional purely photometric descriptors used in images.

5.12. Intrinsic Shape Signatures [H,P,RF,G]

ISS [Zhong 2009] is a point descriptor focused on shape retrieval problems. This method describes a 3D point using two different pieces of information: an LRF based on the eigendecomposition of the neighborhood's covariance matrix and a 3D occupational histogram of the points in its spherical neighborhood.

Given a point set $\mathcal{A}$, a point a_i, and a supporting radius r, the LRF is computed using the eigenvectors of the weighted covariance matrix of the neighborhood of N_{a_i} (e_i^x, e_i^y, e_i^z). Then, a feature vector is computed using a 3D occupational histogram of the supporting neighborhood N_{a_i}. Each neighbor a_k is coded using its polar coordinates with reference to the LRF of a_i. A discrete spherical grid is used to simplify the histogram. Figure 12 shows a graphic example of the process.

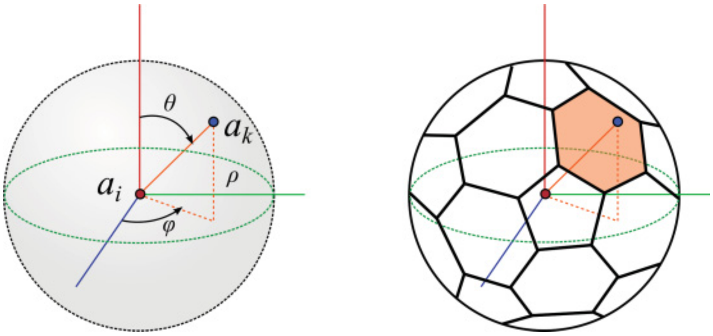

Fig. 12. ISS representation. Left: Polar coordinates for neighboring points (ρ, θ, φ). Right: Spherical grid used to divide the angular space.

The ISS descriptor $f^{\mathrm{ISS}}(a_i)$ is a combination of an LRF of a_i and the 3D shape feature vector. To find the correspondences between two shapes, the authors compare the feature vectors of the candidates using χ^2 statistics to compute the distance between two shape feature vectors.

5.13. Heat Kernel Signature [S,M,nRF,T]

HKS, presented by Sun et al. [2009], is a variation of the heat kernel that is a fundamental solution of the Heat equation. HKS is based on the concept of heat diffusion on a surface over time. The authors propose HKS as both a detector and a descriptor method that possesses many properties: intrinsic, informative, multiscale, and stable against noise and perturbations. To reduce complexity, HKS focuses on the measure of heat diffusion on the considered point alone. The authors use a time parametrization because the time parameter provides a natural notation of scale to describe a shape around a point. The method thereby computes the heat that remains at a certain point at time t. Given a point a_i on a mesh $M_{\mathcal{A}}$, the authors define its HKS ($f^{\mathrm{HKS}}(a_i)$) as a function over the temporal domain, maintaining all the information of the heat kernel:

$$f^{\mathrm{HKS}}(a_i) : \mathbb{R}^+ \to \mathbb{R}, \mathrm{HKS}(a_i, t) = k_t(a_i, a_i). \tag{13}$$

The original heat kernel function $k_t(a_i, a_j) : \mathbb{R}^+ \times M \times M$ (M being a Riemannian manifold) can be interpreted as the amount of heat that is transferred from a_i to a_j in time t given a unique heat source at a_i. Due to the complexity of the computation, the authors restricted the function to a subset of $\mathbb{R}^+ \times M$. Despite this restriction, they showed that the heat kernel function $\{k_t(a_i, a_i)\}_{t<0}$ keeps all information of $\{k_t(a_i, a_j)\}_{t<0}$.

The scale of the descriptor is given at timed intervals. For small values of t, the descriptor is focused on small neighborhoods, which provide a more detailed description. It can be used to describe the curvature of the surface. For large values of t, large neighborhoods are taken into account, obtaining a global descriptor of the shape, distinguishing large parts of the same object. Figure 13 is an example of the performance of the HKS: at small scales the claws are similar to each other. With large values of t, we can distinguish different parts of the dragon like front feet, back feet, head, and tail.

The authors use local maxima of the function $k_t(a_i, a_i)$ for large t. Point a_i is a feature if $k_t(a_i, a_i) > k_t(a_j, a_j)$ for all a_j in the two-ring neighborhood of a_i. The correspondence function cf is basically the comparison between both descriptors $f^{\mathrm{HKS}}(a_i)$ and $f^{\mathrm{HKS}}(b_j)$.

According to Sun et al. [2009], despite the restrictions applied to the heat kernel, HKS preserves all of the shape information, as well as the stability against perturbations.

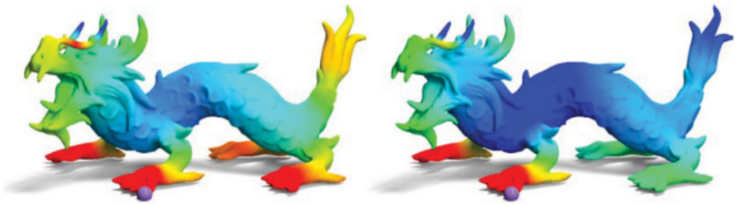

Fig. 13. Color plot of the difference between the HKS defined by the range of scales $[t_1, t_2]$ of the point marked by the purple sphere and the signatures of other points on the shape. The difference increases as the color changes from red via green to blue. Left: Both t1 and t2 are small. Right: t1 is small, whereas t2 is large. Figure taken from Sun et al. [2009].

The main drawbacks are the computation of the eigendecomposition, which is costly. This means that the computing time, in certain conditions with very large point clouds, can prove to be impractical.

Dey et al. [2010] presented an application of HKS for the pose-oblivious matching of incomplete models. HKS is used to obtain a segmentation of the model to perform shape retrieval from a data base of complete, incomplete, or partial models. HKS is used in conjunction with a point selection method based on persistent homology that consists of selecting a subset of the maximum values of HKS across different scales with large topological persistence.

Ovsjanikov et al. [2010] presented a method that uses HKS to conduct matching with isometries using only one-point correspondence on rigid and nonrigid transformations. The authors presented a new approach called *Heat Kernel Maps*. Given a fixed point in a manifold $\mathcal{A}$, this method creates a global shape descriptor. For each point a_i in $\mathcal{A}$, a heat kernel function is computed:

$$\Phi_p^{\mathcal{A}} : \mathcal{A} \rightarrow F, \Phi_p^{\mathcal{A}}(a_i) = k_t^{\mathcal{A}}(p, a_i), \tag{14}$$

where F is the space of functions from $\mathbb{R}^+$ to $\mathbb{R}^+$. Thus, $\Phi_p^{\mathcal{A}}$ associates a real-valued function to every point $a_i \in \mathcal{A}$. This function is a one-parameter function (t) given by $k_t^{\mathcal{A}}(p, a_i)$. The key issue in this approach is that only one correspondence is needed for the matching process. This is the direct consequence of the authors proving formally that the heat kernel map is *injective*.

5.14. Rotational Projection Statistics [S,M,RF,G]

Rotational projection statistics (RoPS) [Guo et al. 2013] is a local feature descriptor that describes a point using a coarse partition of a 2D projection plane with rotational statistics of the surface in combination with a robust LRF that is invariant to clutter and occlusions.

As we can also see in other approaches [Chung et al. 1998; Zhong 2009], a robust LRF is computed by performing an eigendecomposition of the covariance matrix of the neighborhood around a given point. This LRF makes the descriptor invariant to rotation and translation changes. However, the sign ambiguity of the LRF results in a lack of precision.

Given a point a_i from $\mathcal{A}$, only a neighborhood N_{a_i} is considered using a sphere of radius r centered at a_i for the descriptor computation. The neighboring points are rotated at an angle θ_k along the x-axis of the LRF ($N_{a_i}^{\theta_k}$). Then, all points are projected into the xy plane, resulting in a 2D representation of $N_{a_i}^{\theta_k}$. This plane is divided into regular cells, and for each cell, the falling points are counted, producing a distribution matrix D. To make the descriptor more compact, several statistics are computed, such as *moment* and *entropy*, from each distribution matrix. Next, this process is repeated

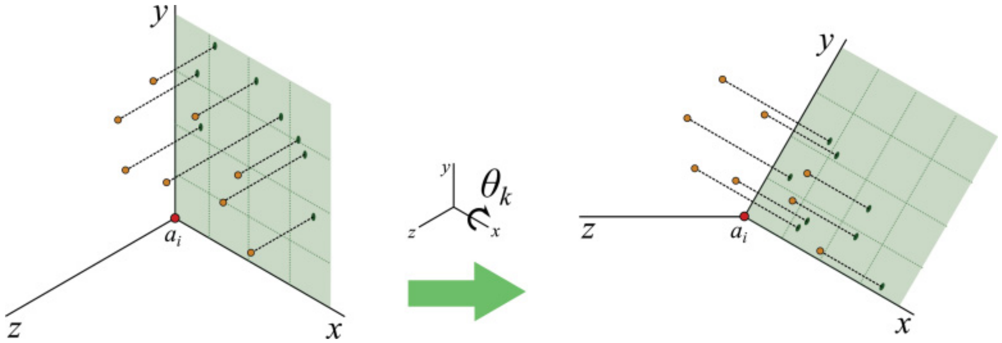

Fig. 14. RoPS representation. Left: Projection of the N_{a_i} into the xy plane from LRF of a_i. This step is done for yz and xz planes as well. Right: Projection of the same N_{a_i} into a rotated xy plane over the x-axis.

by projecting the point cloud into yz and xz planes, obtaining one feature vector for each projection. This operation is repeated with different angles θ_k, $k = 1, 2, \ldots, T$. Finally, all of these steps are repeated for the other local axes y and z. Figure 14 shows a representation of the algorithm. The overall process produces many feature vectors that are concatenated to make the RoPS descriptor.

In Guo et al. [2013], RoPS was compared to other methods such as Spin Image [Johnson and Hebert 1999], local surface patches [Chen and Bhanu 2007], and SHOT [Tombari et al. 2010], obtaining good results in noisy scenes as well as with varying mesh resolutions. However, the tests were made with processed data, taking only 1,000 random feature points of the original models.

6. SEARCHING STRATEGIES

Once the points in the two sets $(\mathcal{A}, \mathcal{B})$ to be matched have been filtered and their shape described, registration algorithms need to find the proper point correspondences between the two sets.

Methods that extract few key points are able to use brute force to find correspondences. However, this process is computationally expensive. Some methods reduce the computation time minimizing the search space, such as 3D Shape Contexts [Körtgen et al. 2003], which preselects the possible candidates satisfying certain criteria and then applies brute force with these candidates. Nevertheless, in most situations, more elaborate algorithms are necessary to report results in a reasonable amount of time.

Considering that at least three points in each set are needed to determine a rigid transformation between two 3D point sets unambiguously, the asymptotic cost of such approaches is in $O(n^6)$. Consequently, the space to be navigated in the search for correspondences is huge. Devising a sophisticated search strategy that is able to take advantage of detection and descriptor information has the potential to greatly reduce computation costs and thus increase the range of application of such registration algorithms. Existing methods implementing searching strategies already achieve very good results in comparison with typical brute-force methods.

As opposed to Fine Matching algorithms, the finality of these searching strategies is to achieve only a rough alignment. The idea of these kinds of methods is to identify the arbitrary position of input shapes and find the transformations between them as quickly as possible. Precision is thus not the most important factor. Instead, robustness is key providing guarantees to subsequent Fine Matching. Henceforth, we describe the most relevant searching strategies in chronological order.

6.1. Algebraic Surface Model

Many methods work with triangulated meshes, or at least with correspondences between points in meshes. Tarel et al. [1998] proposed a Coarse Matching method that estimates a transformation using a polynomial model as a surface representation, without the need for point correspondences. This method consists of creating two polynomial models of the registering surfaces. The authors use a linear algorithm based on least squares called *3L Fitting* to obtain a distance function between the polynomial model and the points of the shape. Unlike other implicit polynomial fitting methods, the linear algorithm does not incur in high computational costs. The only requirement is to have the normal vector of each point in the surface to estimate the model. Furthermore, the computation time is faster than other searching strategies because this method does not need to calculate point correspondences.

However, normal vectors are required to estimate the models. These vectors are not easy to compute. The main drawback of this method is that the overlap between surfaces is required to be high ($\geq$85%). This is not usually the case in real-life applications.

6.2. RANSAC-based Methods

Random Sample and Consensus (RANSAC) is an iterative method designed to find the parameters of a model from a set of data that contains outliers. Given an input noisy data, RANSAC finds the parameters that adjust the input data to a given model, discarding the outliers. This approach is the base of a wide variety of methods. One of them is the approach presented by Chen et al. [1999], which is based on the fact that we can determine a rigid transformation with only three points (a base B). The idea is to find a base in one of the shapes and find the corresponding base in the other shape. The algorithm works as follows. First, determine three different points randomly on the first surface: primary (a_p), secondary (a_s), and auxiliary (a_a). Consider the distances between these three points to be d_{ps}, d_{pa}, and d_{sa}. Each point on the second surface is considered as the corresponding point b_p of the primary point a_p on the first surface. Then, the correspondence of the secondary point is searched on the second surface at distance d_{ps} from b_p. If no point around b_p at distance d_{ps} exists, discard b_p and start again with another primary point on the second surface. However, if there is a secondary point b_s, look for the auxiliary point b_a that satisfies the distances. The transformation between both surfaces can be determined when the base $B_{\mathcal{B}}$ on the second surface is identified. This search is repeated for all bases found. The best transformation is the one with the highest number of corresponding points.

Although this method is robust even with outliers, the main drawback is its computation time. In fact, this method is only usable with a small amount of input data, as was stated in Salvi et al. [2007] and Díez et al. [2012].

Winkelbach et al. [2006] presented an improvement of the classic RANSAC called *Random Sampling* (RANSAM). This approach consists of randomly selecting four oriented points (points with their normal vector) from both surfaces ($(a_i, a_h) \in \mathcal{A}$ and $(b_j, b_k) \in \mathcal{B}$) using a Monte-Carlo algorithm. Bases of two oriented points are sufficient to determine a rigid transformation. This yields a searching complexity of $O(n^2)$.

The correspondences between points are encoded in 4D relation vectors. These vectors consist of the Euclidean distance between the points, the angles of inclination between their normal vectors, the line connecting them, and the rotation angle between the normals around the connection line. The search for correspondences is performed using a hash table that stores these relation vectors. The use of this hash table allows the complexity of the algorithm to drop to $O(n)$.

Whenever a correspondence is found, a rigid transformation is computed. The quality of the registration is measured then by estimating the proportion of the overlapping areas between the two point clouds. A movement is considered a solution if the distance between points in both surfaces is smaller than a certain threshold. The selection of points can be improved using descriptors that weight the random selection.

Another method that uses randomized algorithms is *property testing* [Ron 2001]. This approach consists of determining whether a given object has a predetermined property or is "far" from any object having the property. This methodology can be applied in computational geometry problems [Czumaj et al. 2000].

6.3. Robust Global Registration

Gelfand et al. [2005] presented a Coarse Matching approach based on looking for correspondences using a branch-and-bound algorithm. Given two shapes $\mathcal{A}$ and $\mathcal{B}$, the method consists of extracting a set of key points $S_{\mathcal{A}}$ from $\mathcal{A}$ using an Integral Invariants volume descriptor [Manay et al. 2004]. To increase the robustness, the authors use multiscale resolution in the description process. For each feature point a_i in $S_{\mathcal{A}}$, the algorithm finds a subset of points in $\mathcal{B}$, called $C_{\mathcal{B}}(a_i)$, with a high correspondence with a_i. To reduce the correspondence list of points, a thresholding function is applied over $C_{\mathcal{B}}(a_i)$, considering a pair of points (a_i, a_j) from $S_{\mathcal{A}}$ and a pair of potential corresponding points (b_i, b_j) from $C_{\mathcal{B}}(a_i)$. The distance between a_i and a_j needs to be approximately the same as the distances between their correspondences in the model.

The searching strategy uses a branch-and-bound algorithm that creates a solution-candidate tree where each branch represents one possible correspondence set of points from $\mathcal{B}$. In each level of the tree, one possible candidate is added to the solution. If one of these possible candidates does not pass the threshold test, and thus the RMSD between $\mathcal{A}$ and $\mathcal{B}$ is not improved, the entire branch is pruned. The whole tree is explored finding the best correspondence set where all correspondences pass the threshold and provide the minimum error (RMSD) between both shapes.

This method is robust to noise and works well with occluded scenes and partial registration. However, the algorithm needs strong feature points to obtain good alignments. The uncertainty created due to the use of weak feature points increases the error both between registered points and the searching time.

The authors also extend their algorithm to detect symmetries registering an object with a copy of itself.

6.4. 4-points Congruent Sets

Aiger et al. [2008] presented a searching strategy that takes advantage of the geometric properties of coplanar groups of four points to devise a method that, while using more than the usual three points to determine motions, can be shown to incur lower asymptotic costs. The method finds a transformation between two views using a coplanar set of points with no assumption about the initial alignment.

The authors use four coplanar points from $\mathcal{A}$ to build a base $B_{\mathcal{A}}$ and find its correspondent base in $\mathcal{B}$. Although the extra point is not mandatory to compute a movement, it makes the process more robust and allows the authors to provide proof of reduced asymptotic costs. The key to this method in terms of speed is the use of wide bases. Figure 15 presents a comparison between wide and narrow bases. Performing the alignment process with wide bases makes the registration more robust because the alignment is affected less by errors in accuracy. With narrow bases, a small perturbation of the base might ripple away to induce a noticeable displacement of the full object.

The algorithm works as follows. Given two surfaces $\mathcal{A}$ and $\mathcal{B}$, four almost-coplanar points are selected from $\mathcal{A}$ (base $B_{\mathcal{A}} = \{a_i, a_j, a_k, a_l\}$). The algorithm chooses three random points close to each other and selects the remaining point such that the four points together form a wide base that is approximately coplanar. To find the best

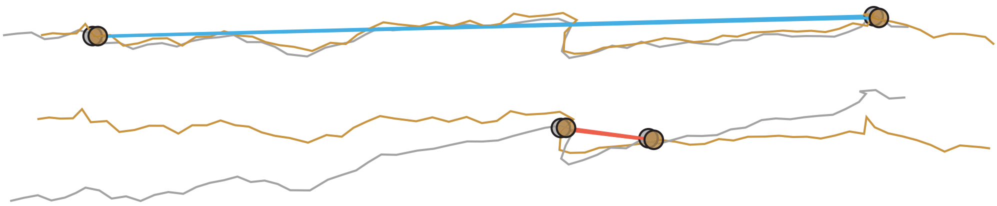

Fig. 15. Comparison between wide base (top) and narrow base (bottom) of the 4PCS method, taken from Aiger et al. [2008]. Golden and gray curves represent two different surfaces to be registered.

four-points set in surface $\mathcal{B}$ that are approximately congruent to $B_{\mathcal{A}}$ (up to an approximation level δ), the authors use a descriptor of *4-points sets*, based on distance ratios between points in the base:

$$r_1 = ||a_i - e||/||a_i - a_j||, \tag{15}$$

$$r_2 = ||a_k - e||/||a_k - a_l||, \tag{16}$$

where e is the intersection point between $a_i a_j$ and $a_k a_l$ lines. These two ratios are invariant under affine transformations. Thus, four-point sets from $\mathcal{B}$ that have approximately the same ratios than $B_{\mathcal{A}}$ are identified. The full algorithm runs in $O(n^2)$.

The authors justify not using local descriptors because they are not robust to noise and outliers, applied with real data. Instead, the authors rely on the principle of large numbers. This principle applies in the sense that although particular point correspondences might be overlooked, the high number of corresponding points between the two sets allow for a large number of solutions. In this approach, this principle requires solving the largest common pointset (LCP) problem. LCP under δ-congruence reports a subset of $\mathcal{B}$ that has the largest possible cardinality, where the distance between $\mathcal{A}$ and $\mu(\mathcal{B})$ is less than δ.

The authors report and provide experimental proof of how the combination of wide bases and LCP makes the registration method resilient to noise and outliers.

The method of 4-points congruent sets (4PCS) is compared with a combination of local descriptors with RANSAC. The authors use Spin Image [Li and Guskov 2005] and Integral Invariants [Pottmann et al. 2009]. As we can see in Figure 16, 4PCS outperforms LD-RANSAC. An additional aspect of 4PCS in this case is that, as opposed to LD-RANSAC, it does not need parameter tuning.

6.5. Evolutionary Methods

Evolutionary methods are searching strategies based on computational models of evolutionary processes that carry out the registration without any initial estimation of the initial alignment and without needing refinement. The idea is to use fitness functions to measure the quality of each potential solution. A remarkable example of the application of genetic algorithms for surface registration can be found in Chow et al. [2004]. Albarelli et al. [2010] presented a game-theoretical approach for surface registration that consists of casting the selection of correspondences in a game-theoretic framework, where a natural selection process allows matching points that satisfy a mutual rigidity constraint to thrive, eliminating all other correspondences. Other strategies are used: stochastic sampling, classical one-point crossover, and simply bit flipping mutation.

A thorough study was presented by Santamaría et al. [2011]. This work reviews the literature concerning evolutionary image registration methods for 3D modeling, including an experimental study. Evolutionary methods are tested against classical ICP methods. One conclusion reached is that the most evolutionary methods outperformed

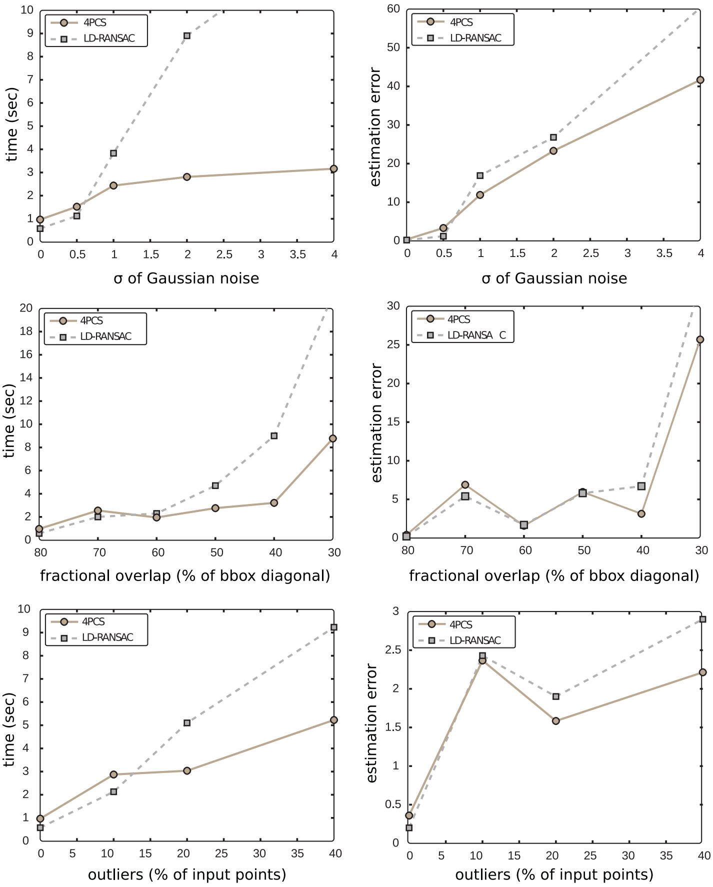

Fig. 16. Performance and comparison taken from Aiger et al. [2008] between 4PCS and LD-RANSAC. We can observe that with low overlap ratios, high level of noise, or many outliers, 4PCS outperforms LD-RANSAC in terms of estimated error and computational time.

the classical approaches based on ICP. Moreover, with EM, a prealignment of input surfaces is not necessary. However, as stated by the authors, genetic algorithms present expensive computational times, making these methods inappropriate when time is a critical factor.

7. REFINEMENT

The last part of the registration process is the refinement of the alignment achieved by Coarse Matching. This step is also commonly referred to as Fine Matching. The most commonly used method today is called *Iterative Closest Point* (ICP), presented by Besl and McKay [1992]. This method has become a standard in the research field of registration due to its robustness and reliability. Given an initial coarse registration, the method associates points from two different point clouds by nearest neighbor criterion, uses mean squared distance minimization functions to estimate the movement, transforms points according to these functions, and iterates until convergence. At the same time as Besl, Yang and Medioni [1992] presented a method following a similar approach. Later, Rusinkiewicz and Levoy [2001] presented several variations of ICP, improving the precision of the algorithm and introducing several filtering methods like

NSS, as mentioned previously. Today, there are many improvements in ICP, such as those achieved by Sharp et al. [2002], Gruen and Akca [2005], Makadia et al. [2006], and Nuchter et al. [2007], to name a few.

Other approaches are able to solve the same problems as ICP, each with a slightly different focus: examples are *Matching signed distance fields* [Masuda 2001, 2002] and *Evolutionary Methods* [Chow et al. 2004], which are in most cases able to solve both the Coarse and Fine Matching problems.

8. DISCUSSION

Every paper studied in this review was tested by the authors under different conditions. For this reason, it is difficult to compare their reported experimental performances. However, there are some reviews and benchmarks that implemented and tested the most commonly used methods [Tangelder and Veltkamp 2004; Salvi et al. 2007; Bronstein et al. 2010; Boyer et al. 2011; Van Kaick et al. 2011; Salti et al. 2011; Dutagaci et al. 2012; Yu et al. 2013; Kim and Hilton 2013; Tam et al. 2013]. Using the results reported in the literature, here we extract some overall conclusions.

To evaluate the performance of the different methods, we focus on three main issues: precision, robustness, and efficiency. We understand *precision* as how accurate the method is, considering error measurements presented in the experimental results of every paper. *Robustness* is the resilience of the method against outside perturbations, such as noise, occlusion, or cluttering. Finally, *efficiency* is measured according to the runtimes provided by the authors, taking into account the data size and the complexity of the algorithm.

Furthermore, we take into account the type of the data used in each proposal. Repositories of processed scanned models like the Stanford Repository,[10] AIM@SHAPE Shape Repository,[11] or Ajmal Mian's Databases[12] are the most widely used because they provide useful data for the tests. However, these models usually consist of only processed data. Besides, there are authors who use scanned data without any preprocessing. This kind of input data provides more realistic situations for the testing of algorithms.

For this discussion, we follow the order of the proposed registration pipeline (Figure 1). We focus on detectors, descriptors, and searching strategies. For a thorough Fine Matching discussion, see Salvi et al. [2007].

8.1. Discussion on Detectors

Detection speeds up computations and thus enhances the range of applicability of algorithms. However, if the detection is not done properly, important information might be lost and existing matches overlooked. If a detection algorithm is able to consistently produce a similar output for the same object under different conditions (noise, change of view, etc.), then this problem is minimized. Consequently, we focus on the repeatability of the detected points over all executions.

A first general conclusion to be drawn is that most of the approaches reviewed use only processed data. Only a few papers present results with real scanned data. Additionally, important differences are observed between these two types of data whenever reported. We believe that obtaining results with real application data represents a mandatory step toward truly practical algorithms.

The methods that achieve higher repeatability results, all with processed data, are Heat Kernel-based Features (HKF) [Sun et al. 2009], Harris 3D [Sipiran and Bustos 2011], and MeshDoG [Zaharescu et al. 2009], which are tested several works [Salvi

[10] http://graphics.stanford.edu/data/3Dscanrep/.

[11] http://shapes.aimatshape.net/.

[12] http://www.csse.uwa.edu.au/.~ajmal.

et al. 2007; Bronstein et al. 2010; Boyer et al. 2011; Salti et al. 2011]. According to Bronstein et al. [2010] and Boyer et al. [2011], HKF achieves approximately 93% of repeatability, retrieving between 9 and 23 feature points from a point cloud of 10,000 points. Under the same conditions, Harris 3D and MeshDoG obtain nearly 83% and 87% of repeatability, respectively. Harris 3D demonstrates notable robustness against holes and topological changes in the input surface, whereas MeshDoG performs when it comes to scaling variances and noise. Furthermore, in Dutagaci et al. [2012], HKF is compared, among other methods, with a human ground truth. These experiments, with processed data alone, demonstrate the good performance of HKF, retrieving points that are usually selected by human subjects.

We mention two other methods, ISS [Zhong 2009] and KPQ [Mian et al. 2010], both tested in Salti et al. [2011]. Although these methods obtain slightly worse results than those of the methods mentioned earlier, both are tested using processed and real data. ISS reports nearly 70% of key-point repeatability using processed data. The authors demonstrate a good recognition range using ISS to retrieve similar models from a database. KPQ achieves lower results in terms of repeatability (≈58%). With real data taken from scans, KPQ achieves similar repeatability results, yet the performance of ISS decreases considerably (≈30%). In terms of temporal efficiency, however, ISS is much faster than KPQ in all tests.

Using volumetric input data, DoG, Harris 3D, and MSV are tested in Yu et al. [2013], among others. The volumetric-specific–designed method MSV obtains the best results yet proves to be slower than the others. In terms of time efficiency, HNSS [Díez et al. 2012] outperforms the runtimes of NSS [Rusinkiewicz and Levoy 2001], obtaining an approximately 99% time reduction in the most extreme case, besides its precision improvements. The authors stress the importance of hierarchical data structures to speed up the searching of correspondences.

Upon analyzing the reported results, we note that there are methods that are more restrictive than others. Specifically, methods like HKF report very few key points, which are expected to convey more distinctive shape information. Conversely, other methods, such as ISS, report many more points in comparison. Although HSF performs better than ISS in most situations, in some others, such as low overlapping ratios or missing data, it might be advisable to opt for more key points. Using fewer feature points in situations where some parts of the shapes are missing might lead to sampling instances that actually prevent the finding of a match. This observation is supported by the results reported in Bronstein et al. [2010], where HKF obtains its worst results when applied to models with holes and shot noise. Consequently, robustness to noise and occlusions should also be considered when choosing which detector to use. Table I summarizes all detectors reviewed in this article.

8.2. Discussion on Descriptors

Descriptors represent a very active field of research in terms of the number of published papers. More precisely, today, object retrieval methods are one of the most popular topics within descriptors. According to the literature, there are many different approaches, but all of them have the same target: providing a useful representation of the shape around the given point that facilitates searching for correspondences between two shapes, thereby avoiding exhaustive searches.

As discussed in Section 5, many of the methods are histogram based. These approaches are easy to implement and also incur low computational costs. However, whenever a part of the compared surfaces is missing or is extremely perturbed, the resulting histogram might also be largely perturbed. This is produced by the strong dependency of these types of descriptors on LRFs. This dependency makes these approaches more sensitive to noise and occlusions. It is not clear whether histogram-based

Table I. Summary of Detector-Based Methods Sorted by Publication Year

Year	Method	Description
2001	**Normal Space Sampling** [Rusinkiewicz and Levoy 2001]	Selects points with high distinctiveness
2006	**Maximally Stable Volumes** [Donoser and Bischof 2006]	Detects stable regions across different scales
2009	**Heat Kernel-based Features** [Sun et al. 2009]	Selects key points according to head distribution function
2009	**MeshDoG** [Zaharescu et al. 2009]	Uses DoG operator to detect points with maximum value of the Laplacian function
2009	**Intrinsic Shape Signatures** [Zhong 2009]	Uses the neighborhood's covariance matrix to detect key points
2010	**Key-Point Quality** [Mian et al. 2010]	Uses the neighborhood's covariance matrix to detect key points
2011	**Harris 3D** [Sipiran and Bustos 2011]	Harris operator for 3D point cloud registration

methods are able to overcome these kinds of problems. However, some methods such as ISS [Zhong 2009], SHOT [Tombari et al. 2010], or ISI [Zhang et al. 2012] show a good resilience to synthetic noise and obtain promising results in many different situations, as we see in Tombari et al. [2010], Salti et al. [2011], and Zhang et al. [2012]. In Tombari et al. [2010], the authors present an experimental study proving that using a unique and unambiguous LRF improves the precision and the accuracy of the descriptor.

In Kim and Hilton [2013], SHOT, Spin Image, Shape Contexts [Körtgen et al. 2003], and FPFH [Rusu et al. 2009] are tested using real data acquired with different techniques. The latter proves to be the most stable and quickest method throughout the tests. With models possessing a high level of irregularities, SHOT works better, yet it fails with regular surfaces.

Another approach to take into account is PCA [Chung et al. 1998]. Although it is not robust to noise and sensible to errors and occlusions, it is one of the fastest methods. For this reason, there are many other algorithms that use PCA as a local descriptor, reporting a local distribution of the neighborhood around a given point. Examples are found in Yang et al. [2006], Pottmann et al. [2009], Sipiran and Bustos [2011], and Darom and Keller [2012], to name a few. Using PCA in a local neighborhood provides descriptions less sensible to noise.

Both in terms of accuracy and time efficiency, the methods reporting the best recent results are Integral Invariants [Manay et al. 2004; Pottmann et al. 2009] and HKS [Sun et al. 2009]. Both methods outperform other algorithms in terms of speed and accuracy, with input data sizes around 100,000 points. Due to the complexity of integral computation, the use of Integral Invariants is slower than HKS. However, most methods are tested only with processed data, where the ground truth is clearly known. With real data, the results are quite different. For example, HKS shows high repeatability and distinctiveness with processed models [Bronstein et al. 2010], yet with laser-scan data or image-based reconstructions, this method is too selective to report a robust registration [Kim and Hilton 2013]. Table II summarizes all descriptors reviewed in this article.

8.3. Discussion on Searching Strategies

There are very few published papers related to searching strategies (Table III). The tendency in the literature is to use good detectors and descriptors, retrieving a number of feature points to use them in brute-force matching strategies. ICP is then used to refine the alignment. Searching strategies make the searching step more efficient

Table II. Summary of Descriptor-Based Methods Sorted by Publication Year

Year	Method	Description	Category	Input Data	Ref. Frame
1996	**Principal Curvature** [Feldmar and Ayache 1996]	Use the principal curvatures of the surface as a descriptor	Sign.	Points	No
1997	**Point Signature** [Chua and Jarvis 1997]	Describe a point with the curvature of the surface arround	Sign.	Points	Yes
1997	**Spin Image** [Johnson 1997]	Histogram of the relative position of the neighbors	Hist.	Mesh	Yes
1998	**PCA** [Chung et al. 1998]	Principal directions of the shape	Sign.	Points	Yes
2003	**3D Shape Contexts** [Körtgen et al. 2003]	Describe a point with the position of certain points of the object	Hist.	Points	No
2003	**Line-based algorithms** [Vanden Wyngaerd and Van Gool 2002]	Describe a point using the curves of the surface	Sign.	Points	No
2006	**Integral Invariants** [Manay et al. 2004]	Descriptor that use the volume below the surface	Sign.	Mesh	Yes
2007	**Point Feature Histograms** [Rusu et al. 2008]	Describe points according to the normal distribution of its neighborhood.	Hist.	Points	Yes
2009	**MeshHOG** [Zaharescu et al. 2009]	Descriptor based on the gradient information over different scales	Hist.	Mesh	Yes
2009	**Intrinsic Shape Signatures** [Zhong 2009]	Histogram of the relative position of the neighbors	Hist.	Mesh	Yes
2009	**Heat Kernel Signature** [Sun et al. 2009]	Descriptor based on the heat diffusion over the surface	Sign.	Mesh	No
2010	**SHOT** [Tombari et al. 2010]	Descriptor that encodes histograms of the normals	Hist.	Mesh	Yes
2012	**Scale-Invariant Spin Images** [Darom and Keller 2012]	Scale-invariant formulation of the spin-image descriptor	Hist.	Mesh	Yes
2012	**Improved Spin Image** [Zhang et al. 2012]	Encode angle information between normals and neighbors	Hist.	Mesh	Yes
2013	**RoPS** [Guo et al. 2013]	Uses rotational statistics of the surface to describe points.	Sign.	Mesh	Yes

Table III. Summary of Searching Strategies Sorted by Publication Year

Year	Method	Description
1998	**Algebraic surface model** [Tarel et al. 1998]	Motion estimation using polynomial models
1998	**RANSAC-based methods** [Chen et al. 1999]	Find the same three-point bases between two models that preserves the Euclidean distances between them
2005	**Robust global registration** [Gelfand et al. 2005]	Find correspondences using a branch-and-bound algorithm
2006	**RANSAM** [Winkelbach et al. 2006]	Select points randomly and use distance and angular relationships between points to find a good movement
2008	**4-points congruent sets** [Aiger et al. 2008]	Use coplanar four-point bases to find the correspondences between two models
2011	**Evolutionary methods** [Santamaría et al. 2011]	Align two different shapes using evolutionary algorithms without no other assumptions

for an initial pose for ICP, helping spread the computational costs while making the process less descriptor dependent.

Concerning robust global registration [Gelfand et al. 2005], the method achieves good results with processed models up to 68,000 points, yet without strong feature points, the computation time and errors increase noticeably. Similarly, evolutionary algorithms [Santamaría et al. 2011] obtain very good results in terms of precision, yet they are computationally expensive dealing with large amounts of data. RANSAM obtains good results in terms of performance and accuracy. Working with nearly 60,000 points, the computation time is reasonable. For example, RMS precisions of 1.03mm, consisting of mechanical ground truth provided with a high-precision turntable, achieves runtime values around 0.5 seconds. No data is provided, however, on the robustness to noise of the algorithm. We feel that this might well prove to be an issue, as algorithms using normal vector information have been reported to present sensibility to noise [Salvi et al. 2007].

Even though no proper experimental comparison of searching strategies exists, we consider that the best approach today is 4PCS [Aiger et al. 2008], because it achieves very good results in many different situations (see Section 6 for details). Using a standard laptop computer, the authors deal with huge point clouds, achieving very accurate registration outputs with low computation time. Moreover, the authors tested their algorithm with real scanned data, achieving more accurate results than others compared to any RANSAC-based method (see Figure 16).

9. CONCLUSIONS

In this article, we reviewed state-of-the-art methods for point cloud rigid registration and proposed a pipelined classification to organize the available approaches.

Working with synthetic or processed data makes it possible to create ground truth values to test the methods. This stands for a far better controlled testing scenario and allows the authors to focus on specific algorithmic aspects. Experimentation with real data, however, is always a crucial step toward application. We believe that the lack of comprehensive studies on real data represents another sign of the intrinsic complexity of the problem.

According to the number of published papers, current trends are focused on detection and description methods, both using the same shape function either to detect or describe the key points. We noted that besides registration between two objects, many approaches also deal with the problem of 3D shape retrieval from object databases. Herein lies an important difference, because these kinds of methods tend to use fewer feature points to reduce the runtime needed to match the model with all objects in the database. The more data points considered, the more precise the final result. Consequently, when working with real data perturbed by noise and occlusions and related to real-life applications, a large number of points is often necessary, and thus efficiency becomes a key factor. Another issue that arises is that most of the studied methods focus on one of the steps of the pipeline only—usually description—and use brute force to determine the final motion. We believe that combining a good descriptor with a sophisticated searching strategy, such as those reviewed in Section 6, would improve the efficiency of the methods. Moreover, this combination has the potential to extend the degree of detail admissible in shape retrieval libraries.

There are few publications focused specifically on improving the searching strategies for matching correspondences between feature points from different shapes. We believe that this field of research may yet produce more efficient methods, using advanced algorithms and data structures such as kd-trees, octrees, or other GPU-friendly structures in this part of the registration pipeline. This fact brings ever closer the registration topic to other fields of research, such as computational geometry, where this problem is tackled from a more theoretical point of view. We believe that the inclusion of these resources has the potential to improve the state of the art, particularly in terms of time efficiency. We believe that this possibility, together with the good performance already shown using real data, shows how searching strategies are ready to be integrated in a fully practical registration pipeline that can deal with a variety of application problems.

ICP and its improvements are the most commonly used refinement methods. The majority of the papers studied in this review used ICP to refine the initial alignments. Although there are other approaches that achieve good results, they incur high computational costs and are only available for small input data.

At this time, it is difficult to compare the performances of existing registration methods. This is due to the lack of a standardized evaluation methodology as well as commonly accepted benchmarks. These results are presented in different magnitudes, or in some cases they go unreported. Although there are some benchmarks [Tangelder and Veltkamp 2004; Salvi et al. 2007; Bronstein et al. 2010; Boyer et al. 2011; Van Kaick et al. 2011; Salti et al. 2011; Dutagaci et al. 2012; Yu et al. 2013; Kim and Hilton 2013; Tam et al. 2013] that provide comparative studies, they are far from comprehensive and difficult to extend to all methods in the literature. We believe that it is necessary to define generic guidelines to be used to test the performances of registration methods. What may be more interesting still might be to define standard conventions for the study and presentation of results. For example, it is important to report the runtime of each part of the process, computer characteristics, data sizes, number of feature points, evaluation measures, and so forth.

To sum up, we believe that most parts of the registration pipeline presented have reached a point where the existing methods have already been shown to be usable in practical situations, or seem to be quite close to it. Consequently, a reasonable expectation is that in a near future, it will be possible to present a registration algorithm optimized in terms of all of these pipeline steps. Such a method would have the potential to greatly increase the current areas of application of point cloud registration, as well as the sizes of the data sets used. We believe that the design of such a method and the provision of experimental proof of its ability to work on a variety of real-life situations represent both a challenge and an opportunity for the research communities involved.

REFERENCES

Dror Aiger, Niloy J. Mitra, and Daniel Cohen-Or. 2008. 4-points congruent sets for robust pairwise surface registration. *ACM Transactions on Graphics* 27, 85.

Andrea Albarelli, Emanuele Rodola, and Andrea Torsello. 2010. A game-theoretic approach to fine surface registration without initial motion estimation. In *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Los Alamitos, CA, 430–437.

Marc Alexa. 2002. Recent advances in mesh morphing. *Computer Graphics Forum* 21, 173–198.

Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. 2011. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proceedings of the 2011 International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE, Los Alamitos, CA, 1626–1633.

Stephen Bailey. 2012. Principal component analysis with noisy and/or missing data. *Publications of the Astronomical Society of the Pacific* 124, 919, 1015–1023.

Paul J. Besl and Neil D. McKay. 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2, 239–256.

Harry Blum. 1967. A transformation for extracting new descriptors of shape. In *Models for the Perception of Speech and Visual Form.* MIT Press, Cambridge, MA, 362–380.

Edmond Boyer, Alexander M. Bronstein, Michael M. Bronstein, Benjamin Bustos, Tal Darom, Radu Horaud, Ingrid Hotz, Yosi Keller, Johannes Keustermans, Artiom Kovnatsky, Roee Litman, Jan Reininghaus, Ivan Sipiran, Dirk Smeets, Paul Suetens, Dirk Vandermeulen, Andrei Zaharescu, and Valentin Zobel. 2011. SHREC 2011: Robust feature detection and description benchmark. In *Proceedings of the 4th Eurographics Conference on 3D Object Retrieval*. 71–78.

Alexander M. Bronstein, Michael M. Bronstein, Benjamin Bustos, Umberto Castellani, Marco Crisani, Bianca Falcidieno, Leonidas Guibas, Iasonas Kokkinos, Vittorio Murino, Maks Ovsjanikov, Giuseppe Patane, Ivan Sipiran, Michela Spagnuolo, and Jian Sun. 2010. SHREC 2010: Robust feature detection and description benchmark. In *Proceedings of the Eurographics 2010 Workshop on 3D Object Retrieval*. 79–86.

Benjamin Bustos, Daniel A. Keim, Dietmar Saupe, Tobias Schreck, and Dejan V. Vranić. 2005. Feature-based similarity search in 3D object databases. *ACM Computing Surveys* 37, 4, 345–387.

Owen Carmichael, Daniel Huber, and Martial Hebert. 1999. Large data sets and confusing scenes in 3-D surface matching and recognition. In *Proceedings of the IEEE International Conference on 3D Digital Imaging and Modeling*. 358–367.

Frederic Cazals and Marc Pouget. 2005. Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometric Design* 22, 2, 121–146.

Chu-Song Chen, Yi-Ping Hung, and Jen-Bo Cheng. 1999. RANSAC-based DARCES: A new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 11, 1229–1234.

Hui Chen and Bir Bhanu. 2007. 3D free-form object recognition in range images using local surface patches. *Pattern Recognition Letters* 28, 10, 1252–1262.

Chi Kin Chow, Hung Tat Tsui, and Tong Lee. 2004. Surface registration using a dynamic genetic algorithm. *Pattern Recognition* 37, 1, 105–117.

Chin Seng Chua and Ray Jarvis. 1997. Point signatures: A new representation for 3D object recognition. *International Journal of Computer Vision* 25, 1, 63–85.

Do Hyun Chung, Il Dong Yun, and Sang Uk Lee. 1998. Registration of multiple-range views using the reverse-calibration technique. *Pattern Recognition* 31, 4, 457–464.

David Cohen-Steiner and Jean-Marie Morvan. 2003. Restricted Delaunay triangulations and normal cycle. In *Proceedings of the ACM Annual Symposium on Computational Geometry*. 312–321.

Nicu D. Cornea, M. Fatih Demirci, Deborah Silver, Ali Shokoufandeh, Sven J. Dickinson, and Paul B. Kantor. 2005. 3D object retrieval using many-to-many matching of curve skeletons. In *Proceedings of the International Conference on Shape Modeling and Applications*. IEEE, Los Alamitos, CA, 366–371.

Nicu D. Cornea, Deborah Silver, and Patrick Min. 2007. Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics* 13, 3, 530–548.

Artur Czumaj, Christian Sohler, and Martin Ziegler. 2000. Property testing in computational geometry. In *Algorithms—ESA 2000*. Lecture Notes in Computer Science, Vol. 1879. Springer, 155–166.

Tal Darom and Yosi Keller. 2012. Scale-invariant features for 3-D mesh models. *IEEE Transactions on Image Processing* 21, 5, 2758–2769.

Tamal K. Dey, Joachim Giesen, and Samrat Goswami. 2003. Shape segmentation and matching with flow discretization. In *Algorithms and Data Structures*. Lecture Notes in Computer Science, Vol. 2748. Springer, 25–36.

Tamal K. Dey, Kuiyu Li, Chuanjiang Luo, Pawas Ranjan, Issam Safa, and Yusu Wang. 2010. Persistent heat signature for pose-oblivious matching of incomplete models. *Computer Graphics Forum* 29, 1545–1554.

Yago Díez, Joan Martí, and Joaquim Salvi. 2012. Hierarchical Normal Space Sampling to speed up point cloud coarse matching. *Pattern Recognition Letters* 33, 2127–2133.

Manfredo P. Do Carmo. 1976. *Differential Geometry of Curves and Surfaces*. Pearson.

Michael Donoser and Horst Bischof. 2006. 3D segmentation by maximally stable volumes (MSVs). In *Proceedings of the 18th International Conference on Pattern Recognition*. 63–66.

Helin Dutagaci, Chun Pan Cheung, and Afzal Godil. 2012. Evaluation of 3D interest point detection techniques via human-generated ground truth. *Visual Computer* 28, 9, 901–917.

Jacques Feldmar and Nicholas Ayache. 1996. Rigid, affine and locally affine registration of free-form surfaces. *International Journal of Computer Vision* 18, 2, 99–119.

Natasha Gelfand, Niloy J. Mitra, Leonidas J. Guibas, and Helmut Pottmann. 2005. Robust global registration. In *Proceedings of the Eurographics Symposium on Geometry Processing*. 197–206.

Armin Gruen and Devrim Akca. 2005. Least squares 3D surface and curve matching. *ISPRS Journal of Photogrammetry and Remote Sensing* 59, 3, 151–174.

Yulan Guo, Ferdous A. Sohel, Mohammed Bennamoun, Jianwei Wan, and Min Lu. 2013. RoPS: A local feature descriptor for 3D rigid objects based on rotational projection statistics. In *Proceedings of the 1st International Conference on Communications, Signal Processing, and Their Applications*. 1–6.

Qi-Xing Huang, Bart Adams, Martin Wicke, and Leonidas J. Guibas. 2008. Non-rigid registration under isometric deformations. *Computer Graphics Forum* 27, 1449–1457.

Natraj Iyer, Subramaniam Jayanti, Kuiyang Lou, Yagnanarayanan Kalyanaraman, and Karthik Ramani. 2005. Three-dimensional shape searching: State-of-the-art review and future trends. *Computer-Aided Design* 37, 5, 509–530.

Andrew E. Johnson. 1997. *Spin-Images: A Representation for 3-D Surface Matching*. Ph.D. Dissertation. Carnegie Mellon University, Pittsburgh, PA.

Andrew E. Johnson and Martial Hebert. 1999. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 5, 433–449.

Kourosh Khoshelham and Sander Oude Elberink. 2012. Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors* 12, 2, 1437–1454.

Hansung Kim and Adrian Hilton. 2013. Evaluation of 3D feature descriptors for multi-modal data registration. In *Proceedings of the 2013 International Conference on 3D Vision*. 119–126.

Iasonas Kokkinos, Michael M. Bronstein, Roee Litman, and Alexander M. Bronstein. 2012. Intrinsic shape context descriptors for deformable shapes. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Los Alamitos, CA, 159–166.

Marcel Körtgen, Gil-Joo Park, Marcin Novotni, and Reinhard Klein. 2003. 3D shape matching with 3D shape contexts. In *Proceedings of the 7th Central European Seminar on Computer Graphics*. 3, 5.

Senthil Kumar, Maha Sallam, and Dmitry Goldgof. 2001. Matching point features under small nonrigid motion. *Pattern Recognition* 34, 12, 2353–2365.

Xinju Li and Igor Guskov. 2005. Multi-scale features for approximate alignment of point-based surfaces. In *Proceedings of the Eurographics Symposium on Geometry Processing*. 217.

Zhouhui Lian, Afzal Godil, Benjamin Bustos, Mohamed Daoudi, Jeroen Hermans, Shun Kawamura, Yukinori Kurita, Guillaume Lavoué, Hien Van Nguyen, Ryutarou Ohbuchi, Yuki Ohkita, Yuya Ohishi, Fatih Porikli, Martin Reuter, Ivan Sipiran, Dirk Smeets, Paul Suetens, Hedi Tabia, and Dirk Vandermeulen. 2012. A comparison of methods for non-rigid 3D shape retrieval. *Pattern Recognition* 46, 1, 449–461.

Yu-Shen Liu and Karthik Ramani. 2009. Robust principal axes determination for point-based shapes using least median of squares. *Computer-Aided Design* 41, 4, 293–305.

Wen Lik Dennis Lui, Titus Jia Jie Tang, Tom Drummond, and Wai Ho Li. 2012. Robust egomotion estimation using ICP in inverse depth coordinates. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, Los Alamitos, CA, 1671–1678.

Ameesh Makadia, Alexander Patterson, and Kostas Daniilidis. 2006. Fully automatic registration of 3D point clouds. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1297–1304.

Siddharth Manay, Byung-Woo Hong, Anthony Yezzi, and Stefano Soatto. 2004. Integral invariant signatures. In *Proceedings of the European Conference on Computer Vision.* 87–99.

Takeshi Masuda. 2001. Generation of geometric model by registration and integration of multiple range images. In *Proceedings of the IEEE International Conference on 3D Digital Imaging and Modeling*. 254–261.

Takeshi Masuda. 2002. Registration and integration of multiple range images by matching signed distance fields for object shape modeling. *Computer Vision and Image Understanding* 87, 1, 51–65.

Takeshi Masuda, Katsuhiko Sakaue, and Naokazu Yokoya. 1996. Registration and integration of multiple range images for 3-D model construction. In *Proceedings of the IEEE International Conference on Pattern Recognition*. 879–883.

Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. 2004. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing* 22, 10, 761–767.

Ajmal Mian, Mohammed Bennamoun, and Robyn Owens. 2010. On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes. *International Journal of Computer Vision* 89, 2, 348–361.

Laurent Najman and Michel Couprie. 2004. Quasilinear algorithm for the component tree. *Proceedings of SPIEVision Geometry XII* 5300, 1, 98–107.

Andreas Nuchter, Kai Lingemann, and Joachim Hertzberg. 2007. Cached kd tree search for ICP algorithms. In *Proceedings of the 6th International Conference on 3-D Digital Imaging and Modeling*. 419–426.

Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. 2012. Functional maps: A flexible representation of maps between shapes. *ACM Transactions on Graphics* 31, 4, 30.

Maks Ovsjanikov, Quentin Mérigot, Facundo Mémoli, and Leonidas Guibas. 2010. One point isometric matching with the heat kernel. *Computer Graphics Forum* 29, 1555–1564.

Karl Pearson. 1901. LIII. On lines and planes of closest fit to systems of points in space. *London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2, 11, 559–572.

Ian R. Porteous. 2001. *Geometric Differentiation: For the Intelligence of Curves and Surfaces*. Cambridge University Press.

Helmut Pottmann, Johannes Wallner, Qi-Xing Huang, and Yong-Liang Yang. 2009. Integral invariants for robust geometry processing. *Computer Aided Geometric Design* 26, 1, 37–60.

Dana Ron. 2001. Property testing. *Combinatorial Optimization Dordrecht* 9, 2, 597–643.

Salvador Ruiz-Correa, Linda G. Shapiro, and Marina Melia. 2001. A new signature-based method for efficient 3-D object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. I–769.

Szymon Rusinkiewicz and Marc Levoy. 2001. Efficient variants of the ICP algorithm. In *Proceedings of the IEEE International Conference on 3D Digital Imaging and Modeling*. 145–152.

Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. 2009. Fast point feature histograms (FPFH) for 3D registration. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 3212–3217.

Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. 2008. Aligning point cloud views using persistent feature histograms. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 3384–3391.

Samuele Salti, Federico Tombari, and Luigi Di Stefano. 2011. A performance evaluation of 3D keypoint detectors. In *Proceedings of the IEEE International Conference on 3D Imaging, Modeling, Processing, Visualization, and Transmission*. 236–243.

Joaquim Salvi, Carles Matabosch, David Fofi, and Josep Forest. 2007. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing* 25, 5, 578–596.

Jose Santamaría, Oscar Cordón, and Sergio Damas. 2011. A comparative study of state-of-the-art evolutionary image registration methods for 3D modeling. *Computer Vision and Image Understanding* 115, 9, 1340–1354.

Gregory C. Sharp, Sang W. Lee, and David K. Wehe. 2002. ICP registration using invariant features. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 1, 90–102.

Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. 2004. The Princeton shape benchmark. In *Proceedings of the Conference on Shape Modeling Applications*. 167–178.

Ivan Sipiran and Benjamin Bustos. 2011. Harris 3D: A robust extension of the Harris operator for interest point detection on 3D meshes. *Visual Computer* 27, 11, 963–976.

Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. 2009. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum* 28, 1383–1392.

Hari Sundar, Deborah Silver, Nikhil Gagvani, and Sven Dickinson. 2003. Skeleton based shape matching and retrieval. In *Proceedings of Shape Modeling International, 2003*. IEEE, Los Alamitos, CA, 130–139.

Andrea Tagliasacchi, Hao Zhang, and Daniel Cohen-Or. 2009. Curve skeleton extraction from incomplete point cloud. *ACM Transactions on Graphics* 28, 3, Article No. 71.

Gary Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank Langbein, Yonghuai Liu, David Marshall, Ralph Martin, Xianfang Sun, and Paul Rosin. 2013. Registration of 3D point clouds and meshes: A survey from rigid to non-rigid. *IEEE Transactions on Visualization and Computer Graphics* 19, 7, 1199–1217.

Johan W. H. Tangelder and Remco C. Veltkamp. 2004. A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications* 39, 3, 441–471.

Jean-Philippe Tarel, Hakan Civi, and David B. Cooper. 1998. Pose estimation of free-form 3D objects without point matching using algebraic surface models. In *Proceedings of the IEEE Workshop on Model-Based 3D Image Analysis*. 13–21.

Federico Tombari, Samuele Salti, and Luigi Di Stefano. 2010. Unique signatures of histograms for local surface description. In *Proceedings of the European Conference on Computer Vision*. 356–369.

Greg Turk and Marc Levoy. 1994. Zippered polygon meshes from range images. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*. 311–318.

Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. 2011. A survey on shape correspondence. *Computer Graphics Forum* 30, 1681–1707.

Joris Vanden Wyngaerd and Luc Van Gool. 2002. Automatic crude patch registration: Toward automatic 3D model building. *Computer Vision and Image Understanding* 87, 1, 8–26.

Simon Winkelbach, Sven Molkenstruck, and Friedrich M. Wahl. 2006. Low-cost laser range scanner and fast surface registration approach. In *Proceedings of the 28th Conference on Pattern Recognition*. 718–728.

Chen Yang and Gérard Medioni. 1992. Object modelling by registration of multiple range images. *Image and Vision Computing* 10, 3, 145–155.

Yong-Liang Yang, Yu-Kun Lai, Shi-Min Hu, and Helmut Pottmann. 2006. Robust principal curvatures on multiple scales. In *Proceedings of the 4th Eurographics Symposium on Geometry Processing*. 223–226.

Tsz-Ho Yu, Oliver J. Woodford, and Roberto Cipolla. 2013. A performance evaluation of volumetric 3D interest point detectors. *International Journal of Computer Vision* 102, 1–3, 180–197.

Andrei Zaharescu, Edmond Boyer, Kiran Varanasi, and Radu Horaud. 2009. Surface feature detection and description with applications to mesh matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 373–380.

Zhiyuan Zhang, Sim Heng Ong, and Kelvin Foong. 2012. Improved spin images for 3D surface matching using signed angles. In *Proceedings of the IEEE International Conference on Image Processing*. 537–540.

Qian Zheng, Andrei Sharf, Andrea Tagliasacchi, Baoquan Chen, Hao Zhang, Alla Sheffer, and Daniel Cohen-Or. 2010. Consensus skeleton for non-rigid space-time registration. *Computer Graphics Forum* 29, 635–644.

Yu Zhong. 2009. Intrinsic shape signatures: A shape descriptor for 3D object recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 689–696.

Received January 2014; revised September 2014; accepted November 2014