Broad-to-Narrow Registration and Identification of 3D Objects in Partially Scanned and Cluttered Point Clouds

Gerasimos Arvanitis, Evangelia I. Zacharaki, Libor Váša, and Konstantinos Moustakas, *Senior Member, IEEE*

***Abstract*—The new generation 3D scanner devices have revolutionized the way information from 3D objects is acquired, making the process of scene capturing and digitization straightforward. However, the effectiveness and robustness of conventional algorithms for real scene analysis are usually deteriorated due to challenging conditions, such as noise, low resolution, and bad perceptual quality. In this work, we present a methodology for identifying and registering partially-scanned and noisy 3D objects, lying in arbitrary positions in a 3D scene, with corresponding high-quality models. The methodology is assessed on point cloud scenes with multiple objects with large missing parts. The proposed approach does not require connectivity information and is thus generic and computationally efficient, thereby facilitating computationally demanding applications, like augmented reality. The main contributions of this work are the introduction of a layered joint registration and indexing scheme of cluttered partial point clouds using a novel multi-scale saliency extraction technique to identify distinctive regions, and an enhanced similarity criterion for object-to-model matching. The processing time of the process is also accelerated through 3D scene segmentation. Comparisons of the proposed methodology with other state-of-the-art approaches highlight its superiority under challenging conditions.**

***Index Terms*—point cloud registration, partially-scanned point clouds, saliency, weighted ICP, cluttered scene.**

I. Introduction

THE scanning and digitization of 3D objects of the real, physical world has recently attracted a lot of attention. Nowadays, there are many applications in different areas (e.g., entertainment, industry, medical visualization, military, heritage, etc.) that utilize 3D objects, either in the form of point clouds or 3D meshes. Future trends show that both this type of applications and the need for reliable 3D object representation will continue to increase. However, in practical scenarios, there are many factors that inevitably affect the quality of the acquired 3D objects, such as illumination conditions or relative motion between device and target during the scanning process, which can create random fluctuation of the data, the formation of additional and unnecessary points on the surface and points away from the surface (outliers). The device itself may also generate a pattern of systematic noise that is added to the surface of the 3D object. Additionally, due to time limitations or a random non-ideal acquisition angle, the point clouds may be incomplete or deformed, which can cause errors in matching and registration [1]. Researchers strive to overcome the existing limitations, trying to provide robust solutions that can be used in realistic circumstances and challenging scenarios. One of the most common research problems upon digitization is the recognition of partially-observed objects in cluttered scenes, which is fundamental in numerous applications of computer vision, such as intelligent surveillance, remote manipulation of robots in manufacturing, autonomous vehicles, automatic assembly, remote sensing, retrieval, automatic object completion. In this work, we assume the existence of scanned point clouds that have been acquired using low-resolution and low-cost 3D scanning devices. These noisy point clouds represent real cluttered scenes consisting of different partially-observed objects, denoted as *query* models. Additionally, we assume the existence of high-quality and complete 3D models, denoted as *target* models, which serve as the ideal representation of the *query* models. The *target* models have been acquired using high-resolution scanning devices, and have also been post-processed to remove noise and outliers. Even though the *query* and *target* models may represent the same object, they have different resolution, orientation, while the *query* object is subject to occlusion, making the processes of matching and registration an arduous task. The objective of this research is threefold;

- To identify different *query* objects partially visible in a point cloud scene.
- To match each *query* object with the corresponding original *target* model (if detected).
- To register *query* and *target* objects in the point cloud scene through 3D registration.

The successful integration of the steps is a challenging task, especially due to the presence of noise and outliers, occlusions, missing parts, different resolutions. Motivated by

Manuscript received November 9, 2020; revised February 23, 2021 and May 14, 2021; accepted June 11, 2021. Date of publication June 17, 2021; date of current version May 11, 2022. This work has received funding from the European Union's Horizon 2020 research and innovation programme under Grant 871738 - CPSoSaware: Crosslayer cognitive optimization tools & methods for the lifecycle support of dependable CPSoS. This project was partially supported by the University Specific Student Research Project SGS-2019-016 Synthesis and Analysis of Geometric and Computing Models. The associate editor coordinating the review of this manuscript and approving it for publication was Lu Fang. *(Corresponding author: Gerasimos Arvanitis.)*

Gerasimos Arvanitis, Evangelia I. Zacharaki, and Konstantinos Moustakas are with the Department of Electrical and Computer Engineering, University of Patras, 26504 Patra, Greece (e-mail: arvanitis@ece.upatras.gr; ezachar@upatras.gr; moustakas@ece.upatras.gr).

Libor Váša is with the Department of Computer Science, and Engineering, Faculty of Applied Sciences, University of West Bohemia, 306 14 Pilsen, Czech Republic (e-mail: lvasa@kiv.zcu.cz).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMM.2021.3089838.

Digital Object Identifier 10.1109/TMM.2021.3089838

the need to overcome all the aforementioned challenges and limitations, inherent in each step of the process, we designed an end-to-end methodology that demonstrates the following main contributions:

- A layered broad-to-narrow registration scheme that reduces the likelihood of getting trapped in local minima, following a RANSAC-style initialization, based on density estimation in the space of rigid transformations, and a subsequent transformation refinement through a novel weighted ICP approach.
- Computational acceleration during object identification through scene segmentation acting as a data selection (i.e. reduction) step.
- Reduction of the effects of noise and object partiality, based on a novel multi-scale saliency extraction technique that allows identification of distinctive regions and reduction of ambiguity in the matching process.
- A novel point cloud descriptor combining pose information with local geometric properties that allows to identify point correspondences even in the case of extreme partiality of the *query* object. An enhanced point cloud similarity criterion is also introduced for accurate target-to-query object matching and registration.

It should be emphasized that our method does not require training data, for the matching process, since it uses only geometric descriptors of each model. Appropriate computational provisions were made in every step of the methodology to produce a fast and robust solution. The rest of this paper is organized as follows: Section 2 presents previous work in related areas. Section 3 describes the proposed method in detail. Section 4 presents the experimental results in comparison with other state-of-the-art methods. Section 5 draws the conclusions and directions for future work.

II. Previous Work

Maybe the most critical step of a 3D object recognition and matching process is the feature descriptor extraction. In the literature, a great number of feature descriptors have been proposed [2], such as spin image [3], direct spacial matching [4], point's fingerprint [5], 3D shape context (3DSC) [6], snapshot [7], local shape descriptors [8], Mesh Histogram of Oriented Gradients (MeshHOG) [9], exponential map [10] and rotational projection statistics [11]. The feature descriptors can be divided into two broad categories: the global feature descriptors and the local feature descriptors. Global feature descriptors represent the geometric and topological properties of the entire 3D model, but they ignore the shape details and require an accurate segmentation of the object. Therefore, they are not usually suitable for the recognition of a partially observed object lying in cluttered scenes. The global feature-based methods are usually used in the context of 3D shape retrieval and classification. Some popular implementations include geometric 3D moments [12], shape distributions [13], viewpoint feature histogram [14], and potential well space embedding [15].

On the other hand, local descriptors focus on narrow neighborhoods, while coarse areas are still present for disambiguation [1]. They can generally handle occlusion and clutter better than the global methods [16], therefore local descriptor approaches are inherently more effective for 3D object recognition [17]. Taati and Greenspan [8] formulated the local shape descriptor for object recognition and localization in range data as an optimization problem. They presented a generalized platform for constructing local shape descriptors that subsumes a large class of existing methods, allowing for tuning to the geometry of specific models. Salti *et al.* [17] developed a hybrid structure between Signatures and Histograms aiming to a more favorable balance between descriptive power and robustness. Their proposed descriptor, called as Signature of Histograms of OrienTations (SHOT), attempts to leverage on the benefits of both Signatures and Histograms approaches. Buch *et al.* [18] introduced a method for fusing several feature matches to provide a significant increase in matching accuracy, which was consistent over all tested datasets. Lu and Wang [1] presented a novel matching algorithm of 3D point clouds based on multiple scale features and covariance matrix descriptors. They applied a combination of the curvature and eigenvalue variation, to precisely detect the key points under multiple scales. Darom and Keller [19] proposed an intrinsic scale detection scheme per interest point and utilized it to derive two scale-invariant local features for mesh models. First, they presented the Scale Invariant Spin Image local descriptor that is a scale-invariant formulation of the Spin Image descriptor, and then, they adapted the SIFT feature to mesh data by representing the vicinity of each interest point as a depth map and estimating its dominant angle using PCA to achieve rotation invariance. Lu *et al.* [2] presented an effective algorithm to recognize 3D objects in point clouds using multi-scale local surface features. It first detects several keypoints in each scene/model and then extracts several feature descriptors with different scales at each keypoint [20], [21].

Point cloud registration refers to the problem of aligning two or more different point clouds that do represent only partially overlapping regions. Generally, the higher the overlap, the easier the registration of the two scenes. However, there are numerous factors that can negatively affect the results [22], such as noise and outliers due to different illumination conditions or relative motion between the scanner and the scene, occlusions and tangled areas. The most well-known and widely used baseline registration method is the Iterative Closest Point (ICP) algorithm. Throughout the years, several ICP-based approaches have been presented [23]–[25] providing very good results; however, the registration is usually inaccurate when the two related point clouds do not have a good relative initial alignment or do not exhibit an ample overlap. Bouaziz *et al.* [26] introduced the sparse ICP which promises superior registration results when dealing with outliers and incomplete data. On one hand, it provides better results than the traditional ICP approach, however, it needs parameter adjustments for different use cases, limiting of application potential. Mavridis *et al.* [27] identified the reasons for the low efficiency of the original Sparse ICP approach and they proposed a registration pipeline that improves the convergence rate of the method by using a more efficient hybrid optimization strategy. Yuan *et al.* [28] try to solve this problem using a combination of ICP and Principal Component Analysis (PCA), which is used to reflect the similarity of the two point clouds. Authors of other works separate the process into a coarse and a fine registration [29], [30]. Moreover, the 4-Points Congruent Sets (4PCS) techniques are very popular, especially for global registration strategies [31]–[33] such as algorithms based on RANdom SAmple Consensus (RANSAC) [34]–[36]. Other related works try to find representative local or global features that are used later as descriptors for matching. Makadia *et al.* [37] used

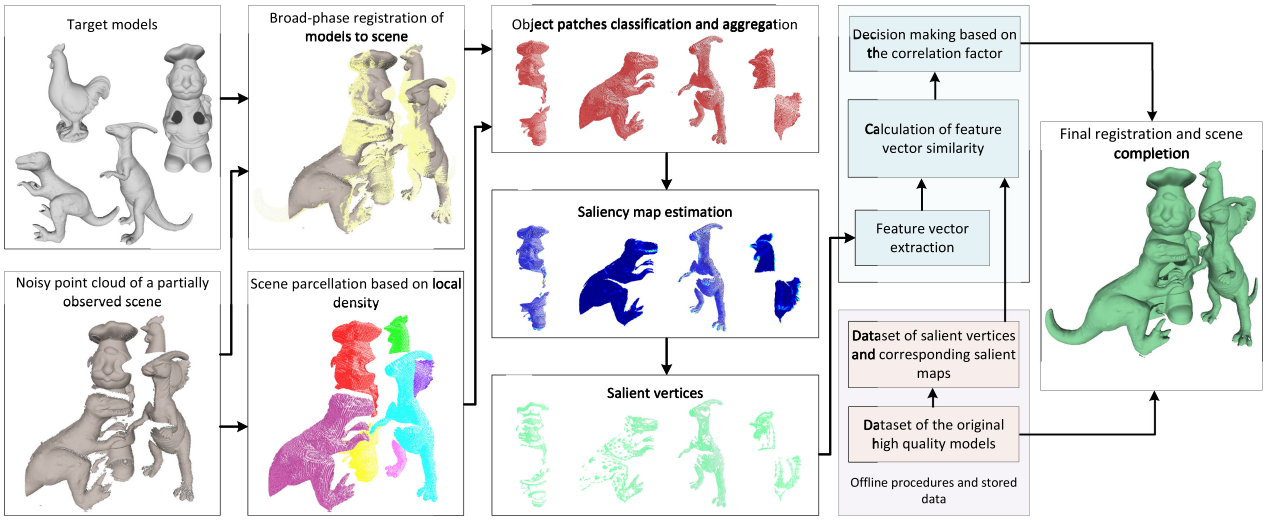

Fig. 1. Pipeline of the proposed methodology. For the sake of simplicity, we illustrate here only the *target* models that are also present in the selected partially observed scene.

extended Gaussian images that can be approximated by spherical histograms of the surface orientations. Their algorithm uses only global information and it does not estimate any local feature. Recently, many deep learning approaches have also been presented [38]–[41], trying to exploit recognizable features from a training dataset, using them later for fast registration, however, they are very vulnerable (as well all the machine learning techniques) in cases where the training and the testing datasets have been created under different circumstances.

III. Work-Flow of the Proposed Method

In this work, we focus on point clouds $\mathbf{P}$ consisting of n vertices $\mathbf{v}$. The i-th vertex $\mathbf{v}_i$ is represented by the Cartesian coordinates, denoted $\mathbf{v}_i = [x_i,\ y_i,\ z_i]^T,\ \forall\ i = 1, \ldots, n$. Thus, all the vertices can be represented as a matrix $\mathbf{V} = [\mathbf{v}_1,\ \mathbf{v}_2,\ \cdots, \mathbf{v}_n] \in \mathbb{R}^{3\times n}$. Let's also denote with Ψ_i^K the set of the K nearest neighbors of point i. Throughout the paper each neighboring point j can be indicated through its vertex coordinates ($\mathbf{v}_j \in \Psi_i$) or, for simplicity, only through its index ($j \in \Psi_i$). Point cloud $\mathbf{P}$ represents the scanned scene, consisting of different partially visible 3D objects. The objective of this work is to match and replace these objects with the corresponding high-quality 3D objects that are assumed to be available beforehand.

The irrelevant objects and the noise seriously affect the optimization process [42]–[50]. Our methodology extends previous work on point cloud registration [34] to overcome its limitations in the case of noisy point clouds. Our idea is that the obtained alignment solution can be improved if registration is guided by the most relevant salient part of the scene. The subsequent steps after the broad-phase registration include feature extraction, similarity assessment and saliency estimation. These steps are more computationally efficient when a part of the point cloud is used rather than the whole scene, as in some cases the scene might be very big. In the next sections, we describe how we robustify and accelerate computations at the same time by identifying and focusing only on salient parts of the scene that potentially correspond to the *target* model. This data selection step not only accelerates the calculations as it allows to search for pairs of landmarks (necessary for the final registration step) in a reduced space, but also leads to a reduction of the impact of the outliers.

An outline of our methodology is illustrated in Fig. 1 and can be summarized as follows:

1) **Broad-phase registration**: First, a fast global registration technique [34] is applied, which helps both for the decision of the matching and the final fine registration process, providing a better initial alignment between the *query* and the *target* object (subsection III-A).
2) **Segmentation, robustification, feature extraction and matching**: In parallel, the whole scene is divided into clusters using our parameter-free implementation of the popular density-based clustering algorithm [51]. Scene clusters, which are geometrically more similar to the registered point cloud of the previous step, are merged to create the *query* object (subsection III-B1). A robustification step is applied to facilitate the identification and removal of spurious point sets (obtained by imperfect scanning) that might blur the object boundaries affecting the registration and the execution time (subsection III-D). The proposed feature vectors, combining pose with local multi-scale geometric information (subsection III-B2), are then extracted and used as descriptors for model to object correspondence assessment (subsection III-B3). Finally, based on the defined point similarity criterion, the best-related pairs of vertices between the matched (complete and partial) objects are identified (subsection III-B4).
3) **Narrow-phase registration**: The final step includes the calculation of a rigid transformation that brings the previously identified pairs of corresponding points into alignment (subsection III-C).

A. Broad-Phase Registration

The first step of the matching process is to align each *target* model to the scanned scene by global registration, without incorporating knowledge of the model class. We have selected a recently proposed global registration algorithm [34] that has shown very good performance in different realistic datasets. For completeness, we present here an overview of the algorithm, while details can be found in [34]. The algorithm finds a number of candidate transformations by matching pairs in a roughly uniformly distributed subset of vertices of the input objects based

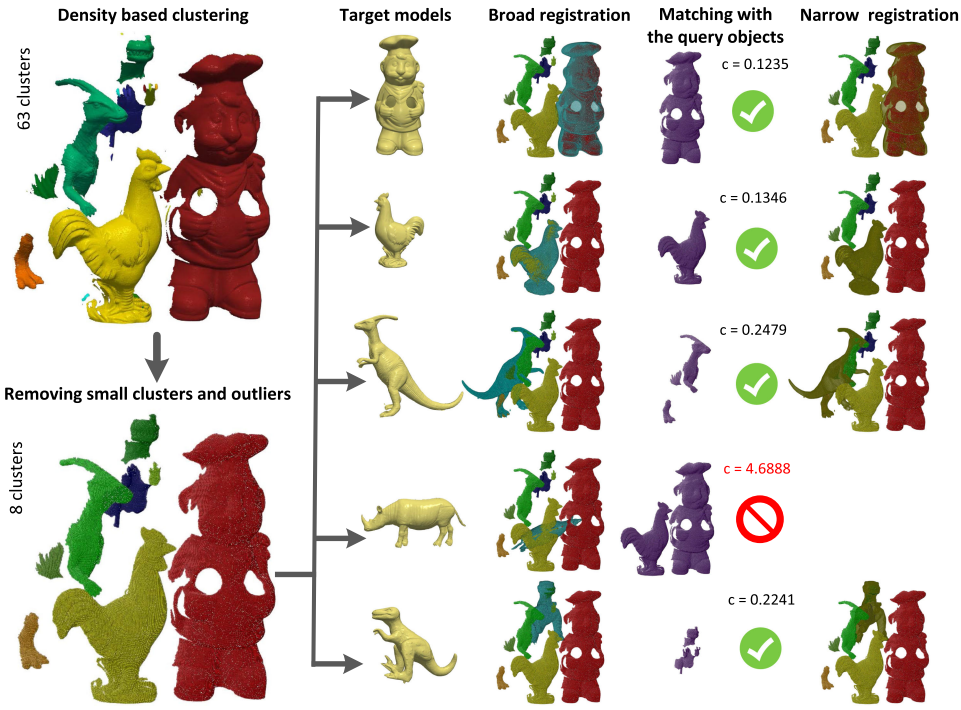

Fig. 2. Broad-to-narrow phase registration. Each target model is registered to the partial scene. The clusters that are geometrically close to the registered model are selected to create the query object for the matching comparison step.

on local shape properties (i.e., principal curvatures and the first principal direction). The optimal transformation is selected by localizing a density peak in the space of candidate rigid transformations. In order to find the density peak, a metric $d(T_1, T_2)$ is needed, which measures the distance of a transformation T_1 from a transformation T_2. A density estimation function can be defined using some kernel function F as:

$$\rho(x) = \sum_i F(d(T_i, x)). \tag{1}$$

Various kernel functions can be used for F, however, mostly for efficiency reasons, a simple Gaussian $F(r) = e^{-(br)^2}$ is preferred, where b is a spread parameter. Instead of looking for the general location of the true global maximum of ρ, ρ is only evaluated at each candidate position and the maximum among them is chosen as the result. Given the spread parameter b and some small threshold ζ, only samples within the radius $r = \sqrt{-ln(\zeta)}/b$ contribute significantly ($> \zeta$) to the density. Therefore, the task is to find, for each candidate, a set of candidates up to the distance r. Because of the non-Euclidean topology of the search space, a KD-tree cannot be used for this purpose, however, a more general acceleration structure - the Vantage Point Tree [52] - can be used. Measuring the distance between two transformations commonly involves relating their rotation and translation components, which is notoriously difficult. Instead, embracing the inherent dependence of such relation on the character of the input data, the metric can be derived from the difference of the effect the two transformations have on the vertices of the input objects:

$$d(T_1, T_2) = \sum_i ||\mathbf{R}_1\mathbf{v}_i + \mathbf{t}_1 - \mathbf{R}_2\mathbf{v}_i - \mathbf{t}_2|| \tag{2}$$

where $\mathbf{R}_1$ and $\mathbf{R}_2$ are the rotations of T_1 and T_2 respectively and $\mathbf{t}_1$ and $\mathbf{t}_2$ are the translation vectors of T_1 and T_2 respectively. Since the sampling density of the input objects may be quite irregular, a more robust option is to integrate distance over triangles instead of summing over vertices. The value of the corresponding integral over a single triangle t can be expressed as:

$$d_t(T_1, T_2) = \int_t ||\mathbf{R}_1\mathbf{v} + \mathbf{t}_1 - \mathbf{R}_2\mathbf{v} - \mathbf{t}_2||da. \tag{3}$$

where a(t) represents the simplest geometric area (i.e., consisting of three vertices) represented by a triangle t. Since this work is designed for point clouds and does not request actual connectivity information, triangles are defined based on the point's closest neighbors. Finally, the full rigid transformation metric is obtained by summing over all triangles:

$$d(T_1, T_2)^2 = \sum_{i=1} d_{t_i}(T_1, T_2)^2 \tag{4}$$

A remarkable property of both expressions for transformation distance, i.e. the vertex sum and the triangle integral, is that with linear pre-processing, they can be evaluated in constant time, i.e. independently on the sampling density of the input objects.

It is important to note here that the *target* object may not appear in the partially observed scene, as presented in the forth line (i.e., rhino model) of Fig. 2. The global registration step is not required to identify if there is a correlation between the *target* model and the *query* model, but only to provide the best possible solution. The final decision on the correspondence of the two models will be made in the narrow-phase registration step according to subsection III-B4.

After the broad-phase registration, all the high-resolution models have been mapped to the scene using global alignment.

Then the solution is further refined to become more robust to outliers.

B. Scene Segmentation and Model-to-Object Matching

1) Point Cloud Segmentation by Density-Based Clustering: The semantic segmentation of the scene is often challenging, as the 3D objects lying in the scene might appear tangled with each other, due to abnormalities created by imperfect scanning. Let's also note that supervised [53] or semi-supervised [54] learning techniques that exploit prior knowledge in the form of shape priors or large training datasets with semantic annotations cannot be applied here to facilitate segmentation, because such large scale annotations are not always available. Our method is based on the assumption that, even if different objects overlap, (i.e. their distance is small in some regions), the local point density within each object is larger than these across different objects in the scene. Therefore for parcellation of the scene, we formulate a density-based algorithm, i.e the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [51], and implement a parameter-free approach, as explained in the sequel. More specifically, DBSCAN is used for the automated segmentation of a point cloud scene into separate clusters, which can be potentially used for matching and registration, reducing the total execution time. The number of clusters is not required beforehand. Considering point cloud $\mathbf{P} \in \mathbb{R}^{n\times 3}$, the clustering result is represented by an vector $\mathbf{k} \in \mathbb{Z}^n$ [55] with elements:

$$\mathbf{k}_i = \begin{cases} j & \text{if } \mathbf{v}_i \text{ belongs to cluster } j \in \{1,\ldots,m\} \\ -1 & \text{if } \mathbf{v}_i \text{ is an outlier} \end{cases}, \quad (5)$$

where m represents the number of clusters. Each scene may consist of a different number of 3D objects and a 3D object may be represented by more than one clusters due to imperfect scanning, occlusions, etc. One parameter of the DBSCAN algorithm which needs to be predefined is the neighborhood radius ϵ. Our contribution is that we allow the threshold ϵ to be spatially adapted and be larger in sparse regions in order to retain sufficient neighbors everywhere. The method becomes more robust with this adaptation, especially in cases where the 3D objects have different density of points in different areas of their surface. More specifically, we assign a value that is inversely proportional to the local density, such as twice the average distance of the K_a nearest neighbors.

$$\epsilon_i = 2\sum_{j=1}^{K_a} \frac{D_{ij}}{K_a} \;\forall\; i = 1,\ldots,n \quad (6)$$

where $D_{ij} \equiv D(i,j)$ is the Euclidean distance of the i vertex to its j^{th} nearest vertex. The spatially adaptive value of ϵ allows us to differentiate which points belong to a cluster, and which are large-scale outliers or noise points. After the broad-phase registration step, all clusters being geometrically close to the registered target model (according to a predefined distance threshold) are merged to form a new point cloud (i.e., *query* object), denoted as $\mathbf{Q}$.

2) Salient Points Detection: Our purpose in this step is to identify if each high-quality *target* model $\mathbf{T} \in \mathbb{R}^{n_t\times 3}$ and each segmented *query* object $\mathbf{Q} \in \mathbb{R}^{n_q\times 3}$ (where $n_t \geq n_q$ due to occlusion, low-quality, etc), represents the same structure. To define similarity between each set of point clouds we propose descriptors that encode spectral saliency. In the following we describe the proposed features, and how they are used to extract point-to-point correspondences, necessary for the final registration step.

The feature descriptors that we use are related to the saliency map of the point cloud. Saliency is a value assigned to each vertex of a point cloud that represents its perceived importance. In the case of raw point clouds without context information, saliency characterizes the geometric properties. High values of saliency represent more perceptually protruding vertices. In this work, we assume that geometric lines, corners, and edges are more distinctive perceptually than flat areas, according to the theory of visual saliency of sight.

To estimate the saliency map, we use a similar pipeline, as the one described in [56], but we extract the saliency map using only spectral analysis, avoiding the computationally complex geometric analysis. For each point $\mathbf{v}_i$ of the point cloud $\mathbf{Q} \in \mathbb{R}^{n_q\times 3}$, we construct a matrix $\mathbf{N}_i \in \mathbb{R}^{(k+1)\times 3}$ comprising of the normals of $\mathbf{v}_i$ and the normals of the k-nearest neighboring points of $\mathbf{v}_i$ (generally, we set $k = 20$):

$$\mathbf{N}_i = [\mathbf{n}_i,\ \mathbf{n}_{i_1},\ \mathbf{n}_{i_2},\ \cdots,\ \mathbf{n}_{i_k}]^T \;\forall\; i = 1,\ldots,n_q \quad (7)$$

For the estimation of the point normals, a plane is approximated based on the set of closest neighboring points, according to [57]:

$$\mathbf{n}_i = \frac{1}{|\Psi_i|}\sum_{\mathbf{v}\in\Psi_i} (\mathbf{v}-\mathbf{v}_i)(\mathbf{v}-\mathbf{v}_i)^T \quad (8)$$

The eigenvector corresponding to the smallest eigenvalue of $\mathbf{n}_i$, is the best estimation of its normal vector. The matrices $\mathbf{N}_i$ are also used for the computation of covariance matrices $\mathbf{C}_i$:

$$\mathbf{C}_i = \mathbf{N}_i^T\mathbf{N}_i \in \mathbb{R}^{3\times 3} \quad (9)$$

The matrix $\mathbf{C}_i = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ is decomposed into a matrix $\mathbf{U}$, consisting of the eigenvectors, and a diagonal matrix $\mathbf{\Lambda} = \text{diag}(\lambda_{i1}, \lambda_{i2}, \lambda_{i3})$, consisting of the corresponding eigenvalues. Finally, the saliency value s_i of a vertex $\mathbf{v}_i$ is determined as the value given by the inverse l_2-norm of the corresponding eigenvalues:

$$s_i = \frac{1}{\sqrt{\lambda_{i1}^2 + \lambda_{i2}^2 + \lambda_{i3}^2}} \quad (10)$$

Based on this equation, flat areas producing high eigenvalues correspond to small saliency values, while the most salient vertices are those that represent high-frequency features (i.e., edges and corners) and thereby produce small eigenvalues. These features are more recognizable by the human brain, since they are perceptually more distinctive. The same process for extracting the saliency maps is followed for partial objects as well as the *target* models. The only difference is that computations for the *target* models are performed only once (offline) and a small set of salient points is stored to be used for the subsequent matching process.

3) Multi-Scale Feature Extraction: First, the saliency values of the two compared models are normalized according to:

$$\begin{aligned} s_{t_i} &= 1 - \frac{e^{-K_s * s_{t_i}}}{s_{\max}} \;\forall\; i = 1,\ldots,n_t \\ s_{q_i} &= 1 - \frac{e^{-K_s * s_{q_i}}}{s_{\max}} \;\forall\; i = 1,\ldots,n_q \end{aligned} \quad (11)$$

where $s_{\max} = \max(\max(s_t), \max(s_q))$. Then, we perform spatial smoothing of the saliency map with a uniform kernel of increasing size and use the obtained values to form a feature vector with the multi-scale saliency values. The neighborhood

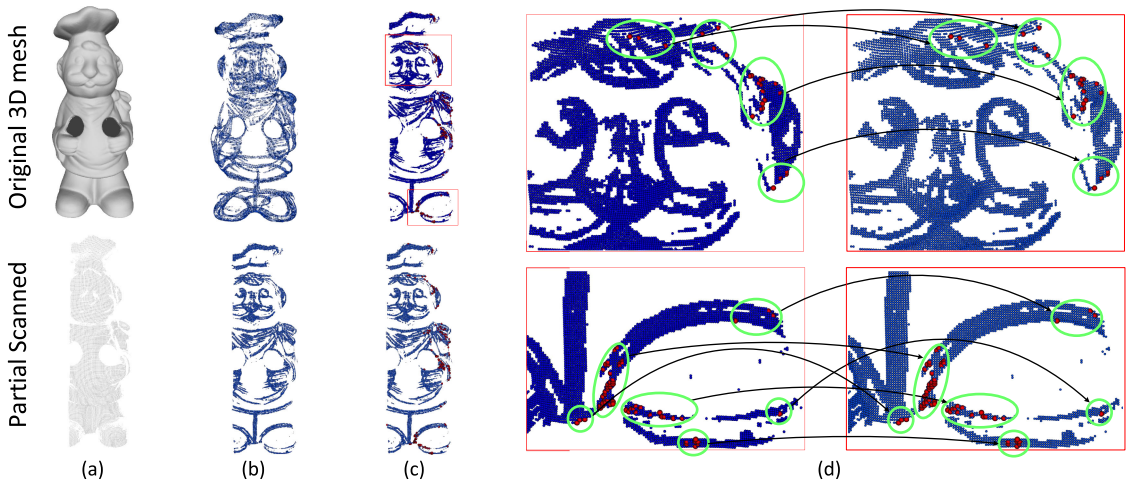

Fig. 3. (a) [Up] Original high-quality 3D mesh (Chef model) (consisting of 176912 vertices), [Down] partially scanned point cloud object of the same model (consisting of 45811 vertices), (b) [Up] salient vertices of the original model (consisting of 55935 vertices), [Down] salient vertices of the partial scanned object (consisting of 16324), (c) [Up] remaining salient vertices of the high-quality model that creates unique pairs with the salient vertices of partially-scanned model (consisting of 11162 vertices) and [Down] vice versa, and (d) Enlarged regions illustrating the identified correspondences in red color.

size is selected as $\Psi^{k \cdot K}$ with $K = 5$ and $k = 1, \ldots, 5$, although these parameters may vary. Smaller scales increase feature vector specificity, while larger scales smooth out noise and irregularities making the shape descriptor more robust. The use of multiple scales allows us to combine both properties and leads to unique and accurate correspondences. This process is applied for each point cloud in $\mathbf{T}$ as well as $\mathbf{Q}$. Specifically, for each vertex i, we create a corresponding vector $\mathbf{a}_i \in \mathbb{R}^5$, according to:

$$\mathbf{a}_i = [\frac{\sum_{j \in \Psi_i^K} s_j}{K} \quad \frac{\sum_{j \in \Psi_i^{2\tilde{}K}} s_j}{2\tilde{}K} \quad \ldots \quad \frac{\sum_{j \in \Psi_i^{5\tilde{}K}} s_j}{5\tilde{}K}]^T \tag{12}$$

Then, we concatenate the multi-scale saliency values with the vertex coordinates to obtain the final feature representation. Finally, for each one vertex, we create the augmented feature vector $\mathbf{f} \in \mathbb{R}^8$, consisting of the vertex coordinates and the corresponding values of the vector $\mathbf{a}$:

$$\mathbf{f} = \mathbf{v} \cup \mathbf{a} = \begin{bmatrix} \mathbf{v}^T \\ \mathbf{a}^T \end{bmatrix} \tag{13}$$

4) Model-to-Object Correspondence Estimation: The feature vectors $\mathbf{f}$, calculated using Eq. (13), are used for the evaluation of similarity between the *query* and *target* point clouds, looking for their unique pairs of vertices $\mathbf{p}$ which exhibit the smallest feature vector distance. It is expressed through the l_2-norm:

$$\mathbf{p} = (\mathbf{v}_t, \mathbf{v}_q) = \arg\min_{\mathbf{v}_t, \mathbf{v}_q} ||\mathbf{f}_t - \mathbf{f}_q||_2 \tag{14}$$

Finally, we keep only the first K_p pairs having the highest feature vector similarity. An example is shown in Fig. 3 c-3 d, where feature vector correspondences are illustrated with red color. These K_p pairs are the best-identified correspondences between model $\mathbf{T}$ and object $\mathbf{Q}$. Let's note here that we use the augmented feature vectors $\mathbf{f}$, which include multi-scale geometric descriptors in addition to 3D location, to avoid erroneous surface mapping obtained by chance due to accidentally good local geometric fit (small Euclidean distance) of the partial point cloud. These augmented feature vectors can ensure not only spatial proximity, but also local shape similarity.

At this point of the methodology, the obtained pairs of corresponding vertices include matches for each *target* model $\mathbf{T}$ to each *query* object $\mathbf{Q}^j, j \in \{1, \ldots, m_q\}$, where the m_q denotes the number of the *query* objects. To identify the correspondences, we introduce a dissimilarity factor c_j which is defined as the mean distance of the K_p pairs of the best-related vertices between model $\mathbf{T}$ and each object $\mathbf{Q}^j$.

$$c_j = \frac{1}{K_p} \sum_{i=1}^{K_p} ||\mathbf{f}_{t_i} - \mathbf{f}_{q_i^j}||_2 \tag{15}$$

where $\mathbf{f}_{q_i^j} \in \mathbf{Q}^j$ is the feature vector matched to $\mathbf{f}_{t_i} \in \mathbf{T}$. The lower the value of c, the more similar are the two point clouds.

C. Narrow-Phase Registration

Having identified and matched the target and query object pairs in the scene, the fine registration is achieved by identifying a set of corresponding points and then finding the optimal transformation that brings those pairs of points (control points) into alignment. In this step, we initialize the registration with the solution obtained from the global initial alignment and refine it using a weighted ICP approach. The objective is, given a set of control points $\mathbf{p} = (\mathbf{v}_{t_i}, \mathbf{v}_{q_i})$ with $\mathbf{v}_t \in \mathbf{T}$ and $\mathbf{v}_q \in \mathbf{Q}$, to estimate a rigid transformation T that minimizes a distance (or more general an error) function.

As the partial object contains only a subset of the shape represented by the high resolution model, point matching is performed starting with a set of control points in $\mathbf{Q}$ and identifying the corresponding points in $\mathbf{T}$ as described in the previous section. Generally, in weighted ICP approaches [58], [59] the error function is composed of a feature vector distance term and a weighting term used to downgrade the contribution of pairs that have high likelihood to be outliers or wrong correspondences. The optimal transformation is obtained by solving a weighted least squares minimization problem. For rigid transformations

expressed by a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$, it can be written as:

$$\arg\min_{\mathbf{R},\mathbf{t}} \sum_i \phi(D(\mathbf{R}\mathbf{v}_{t_i} + \mathbf{t}, \mathbf{v}_{q_i})) \tag{16}$$

where D is the Euclidean distance and $\phi(.)$ is an even, C^1-continuous on $\mathbb{R}$ and monotonically increasing function on $[0, \infty)$. The function $\phi(r)$ that we use is the Tukey's bi-weight function formulated as:

$$\phi(r) = \begin{cases} \frac{\gamma^2}{6}\{1 - (1 - \frac{r^2}{\gamma^2})^3\} & \text{if } |r| \leq \gamma \\ \frac{\gamma^2}{6} & \text{if } |r| > \gamma \end{cases} \tag{17}$$

and we use as weights $w(r)$, the first order derivative of $\phi(r)$ function, denoted $w(r) = \phi'(r)$:

$$w(r) = \begin{cases} r(1 - \frac{r^2}{\gamma^2})^2 & \text{if } |r| \leq \gamma \\ 0 & \text{if } |r| > \gamma \end{cases} \tag{18}$$

D. Robustification by Outliers Removal

Outliers and other surface abnormalities of the scanning procedure might be interpreted as salient points, leading to wrong matching and registration results. So, we have to ignore both the outliers and the open edges, which form the boundary of the segmented point cloud, because they do not represent characteristic discriminative features to guide the registration process.

1) Small-Scale Outliers Removal: Scanned objects or scenes usually include noisy parts represented by vertices that do not belong to the geometry of the real object. Two different types of outliers occur in scanned point clouds; (a) the large-scale outliers that lie away from the point cloud and (b) the small-scale outliers which are tangled with the useful information and could be mistakenly recognized as points [60]. As we mentioned earlier, the large-scale outliers could be removed through the application of the clustering method presented in Section III-B1. For the small-scale outliers removal process, we use a Robust Principal Component Analysis (RPCA) approach, which decomposes the observed measurement matrix $\mathbf{E}$ into a low-rank matrix $\mathbf{L}$, representing the real data, and a sparse matrix $\mathbf{S}$, representing the outliers [61], [62], by solving:

$$\arg\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1, \quad \text{subject to } \mathbf{L} + \mathbf{S} = \mathbf{E}, \tag{19}$$

where $\|\mathbf{L}\|_*$ is the nuclear norm of a matrix $\mathbf{L}$ (i.e, $\sum_i \sigma_i(\mathbf{L})$ is the sum of the singular values of $\mathbf{L}$). This problem can be solved using a splitting method, such as the Augmented Lagrange Multiplier (ALM) algorithm [63], [64]:

$$\arg\min_{\mathbf{L},\mathbf{S},\mathbf{Y}} \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 + \langle \mathbf{Y}, \mathbf{E} - \mathbf{L} - \mathbf{S}\rangle + \frac{\mu}{2}\|\mathbf{E} - \mathbf{L} - \mathbf{S}\|_F^2 \tag{20}$$

where $\langle \mathbf{a}, \mathbf{b}\rangle$ represents the inner product of $\mathbf{a}$ and $\mathbf{b}$. The observed matrix $\mathbf{E} \in \mathbb{R}^{3n\times(k+1)}$ is constructed as follows:

$$\mathbf{E} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_{11} & \mathbf{v}_{12} & \dots & \mathbf{v}_{1^{\frown}k} \\ \mathbf{v}_2 & \mathbf{v}_{21} & \mathbf{v}_{22} & \dots & \mathbf{v}_{2^{\frown}k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{v}_{n_q} & \mathbf{v}_{n_q1} & \mathbf{v}_{n_q2} & \dots & \mathbf{v}_{n_qk} \end{bmatrix} \tag{21}$$

where $\mathbf{v}_i = [x_i, y_i, z_i]^T$ and k represents the number of the nearest neighbors used ($k = 50$). The estimation of the low-rank and the sparse matrix is performed according to:

$$\begin{matrix} \min_{\mathbf{S}} \mathcal{L}(\mathbf{L}, \mathbf{S}, \mathbf{Y}) = \mathcal{Q}_{\lambda\mu^{-1}}(\mathbf{E} - \mathbf{L} + \mu^{-1}\mathbf{Y}) \\ \min_{\mathbf{L}} \mathcal{L}(\mathbf{L}, \mathbf{S}, \mathbf{Y}) = \mathcal{D}_{\mu^{-1}}(\mathbf{E} - \mathbf{S} + \mu^{-1}\mathbf{Y}) \end{matrix} \tag{22}$$

where $\mathcal{Q}_\tau[.] = \text{sgn}(.)\max(|.| - \tau, 0)$ denotes the shrinkage operator and $\mathcal{D}_\tau(.) = U\mathcal{Q}_\tau(\sum)V^T$ denotes the singular value thresholding operator. More details about the RPCA can be found in [63], [64]. The resulting sparse matrix $\mathbf{S} \in \mathbb{R}^{3n_q\times(k+1)}$ has the form:

$$\mathbf{S} = \begin{bmatrix} \tilde{\mathbf{v}}_1 & \tilde{\mathbf{v}}_{11} & \tilde{\mathbf{v}}_{12} & \dots & \tilde{\mathbf{v}}_{1^{\frown}k} \\ \tilde{\mathbf{v}}_2 & \tilde{\mathbf{v}}_{21} & \tilde{\mathbf{v}}_{22} & \dots & \tilde{\mathbf{v}}_{2^{\frown}k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{v}}_{n_q} & \tilde{\mathbf{v}}_{n_q1} & \tilde{\mathbf{v}}_{n_q2} & \dots & \tilde{\mathbf{v}}_{n_qk} \end{bmatrix} \tag{23}$$

where $\tilde{\mathbf{v}}_i = [\tilde{x}_i, \tilde{y}_i, \tilde{z}_i]^T$ represents the corresponding sparse values. We use only the values $\tilde{\mathbf{v}}_i$ of the first column in order to estimate the metric m_i, according to the following equation.

$$m_i = \sqrt{\tilde{x}_i^2 + \tilde{y}_i^2 + \tilde{z}_i^2} \tag{24}$$

The value m_i quantifies the probability of vertex i to be an outlier. The sparse matrix is usually full of zeros (representing vertices on flat areas), and only some high non-zero values exist that correspond to outliers.

2) Boundary Edges Identification and Removal: Similarly to the outliers, the boundary edges of a partially-scanned object may mistakenly be recognized as edge features, so they must be identified and removed as well. It is necessary to differentiate salient points on edges and corners, useful for guiding the registration process, from misleading boundary edges around holes and missing parts, caused by imperfect scanning. To identify such boundary edges, we estimate the mean distance $d_{\Psi_i} = \frac{\sum_{\mathbf{v}\in\Psi_i^{K_D}} \|\mathbf{v} - \mathbf{v}_i\|}{K_D}$ between each i vertex and its K_D nearest neighbors, and characterize each point as internal or boundary point according to:

$$\mathbf{v}_i = \begin{cases} \text{is an internal vertex, if } d_{\Psi_i} \leq 2\bar{d} \\ \text{is a boundary vertex, if } d_{\Psi_i} \geq 2\bar{d} \end{cases} \tag{25}$$

where the $\bar{d}$ represents the mean distance of all mean distances d_{Ψ_j} in the *query* object $\mathbf{Q}$,

$$\bar{d} = \frac{\sum_{i=1}^{n_q} d_{\Psi_j}}{n_q} \tag{26}$$

IV. Experimental Analysis

A. Experimental Setup, Datasets and Metrics

The experiments were carried out on an Intel Core i7-4790HQ CPU @ 3.60 GHz PC with 16 GB of RAM. The core algorithms are written in Matlab and C++.

For the experiments, we used two different datasets. The first one consists of a variety of partially scanned point clouds representing cluttered scenes of different objects, denoted as UWAOR [65], [66]. The dataset contains 50 cluttered scenes with up to 5 objects acquired with the Minolta Vivid 910 scanner in various configurations from a single viewpoint. All objects are heavily occluded (60% to 90%), as illustrated in Fig. 4. The

Fig. 4. UWAOR dataset consisting of 50 partially scanned scenes from a single viewpoint.

Fig. 5. UWA3M dataset consisting of incomplete scans of 3D models in arbitrary angles.

second dataset, Fig. 5, consists of a variety of incomplete models under different viewpoints (angles), denoted as UWA3M [67] and with various percentages of occlusion. There are 22 incomplete instances of "Chef," 16 of "Chicken" and "Parasaurolophus," and 21 of "T-rex". The third dataset [68] is composed of 150 synthetic scenes, captured with a (perspective) virtual camera, and each scene contains 3 to 5 objects. The model set is composed of 20 different target objects.

An index of performance of the registration task is the degree of deviation for rotation $\mathbf{R}$ and translation $\mathbf{t}$ from the ground truth rotation matrix $\mathbf{R}_g$ and translation vector $\mathbf{t}_g$. The rotation error ϵ_r and the translation error ϵ_t are defined as:

$$\epsilon_r = \arccos\left(\frac{\mathcal{S}(\mathbf{R}\mathbf{R}_g^{-1}) - 1}{2}\right)\frac{180}{\pi} \tag{27}$$

$$\epsilon_t = ||\mathbf{t} - \mathbf{t}_g|| \tag{28}$$

where $\mathcal{S}(\mathbf{A})$ determines the sum of the diagonal elements of the matrix $\mathbf{A}$ [1]. It is obvious that the smaller these errors, the better the matching results.

B. Parameter Adjustment

In this paragraph, we will present and justify the selection of parameter values that are fixed through the steps of the proposed methodology in order to provide reproducible results. The chosen number of neighboring vertices in Eq. (6) is equal to $K_a = 5$, but in any case, we have observed that the algorithm is not sensitive to this value. In fact, the results are very similar for the range of $K_a \in [5, 8]$. The only important fact is that those vertices should be retrieved from the geometric area, which is very close to the reference vertex, such as the first ring area.

In Eq. (24), we estimate the quantity m_i for each i vertex that is used to identify if a vertex is an outlier or not. However, we first need to specify a threshold for this identification. We interpret as outliers those vertices that have a value of m_i bigger than 0.6, and then we remove them. Obviously, this threshold can be adjusted; The experimental analysis demonstrates that the lower the threshold, the more vertices are considered as outliers and are therefore removed. The user can adjust the value of this parameter; however, we suggest the threshold to be equal to 0.6.

Eq. (25) is used to determine if a vertex is considered as a boundary point or not. The used number of neighboring vertices in this case is $K_D = 10$. The selection of this value is not critical and it is advised to be selected in the range of $K_D \in [8, 12]$. These values provide enough, but not too many, instances. In subsection III-B4, we described the use of feature vectors $\mathbf{f}$ for finding unique pairs. However, to make the process more computationally efficient, we do not compare the two full-sized point clouds, vertex by vertex, but we keep only the highest saliency vertices (as presented in Fig. 3-(b)), corresponding to saliency values higher than a threshold set to $\tau = 0.4$.

Eq. (15) expresses the dissimilarity factor c. If the value of c is lower than a threshold then we assume that the two examined point clouds are related, representing the same 3D object. The experimental analysis (please refer to subsection IV-D1) has shown that a threshold $\xi = 0.4$ allows to differentiate intra- from inter-class object pairs. We also define the number of the compared vertices to be equal to $K_p = 200$.

We would like to mention here that the selected values are invariant to affine transformations and sampling density. Table III summarizes the default values that we used and a short description.

C. Computational Efficiency

In this paragraph, we present a time complexity analysis showing how our contribution can speed up the clustering process. Table I assesses the execution time of the narrow-phase registration phase and Table II assesses the modified clustering algorithm.

More specifically, in Table I, we present the speed up of our approach in comparison with the one-shot registration (i.e, O. Registration in Table I) (without scene segmentation) applied for different *target* to scene registration examples. In this case, the effectiveness of our approach depends mainly on the size of the *query* point cloud, however for all tested examples the speed up is more than 5x. We would like to note here that the pre-processing step consists of the outliers' removal process and feature extraction.

In Table II, we present the execution times of the original DBSCAN algorithm and our approach for some random scenes of the used dataset (UWAOR). As we can see, our approach is up to 97.7 times faster. Also, we can observe that the effectiveness of our approach is more apparent with increasing number of points. The main reason why our implementation is faster than the original DBSCAN is that our approach does not exhaustively estimate distances between each vertex and all the other vertices of the point cloud, but searches only within a small spatial area consisting of a specific and predefined number of neighbors equal to K_a. In this case the time complexity of the algorithm is not $O(n^2)$ but $O(K_a^2)$, where $K_a \ll n$.

D. Performance Evaluation

In the following subsection, we present and evaluate the performance and accuracy of the proposed matching and registration process.

1) Evaluation of Matching in Partial Scenes: For this experiment, we used the 50 partially scanned scenes of the UWAOR dataset (Fig. 4) and the five *target* models (i.e., Chef, Chicken, Parasaurodophus, T-rex, Rhino). Fig. 7 presents the boxplots of the dissimilarity factor between the *query* and the *target* models.

TABLE I
SPEED UP OF OUR APPROACH (NARROW-PHASE REGISTRATION)

	O. Registration (in sec.)			Our Approach (in sec.)				
Target to scene	**Pre-processing**	**Registration**	**Total**	**Segmentation**	**Pre-processing**	**Registration**	**Total**	**Speed up**
Chef to Scene 5	26.08	9.82	35.90	0.81	3.78	1.81	6.40	**x 5.6**
Chic. to Scene 12	27.91	12.80	40.71	0.29	2.29	2.15	4.73	**x 8.6**
Par. to Scene 27	29.36	26.82	56.18	0.31	4.20	5.08	9.59	**x 5.8**
T-rex to Scene 32	26.72	18.48	45.20	0.16	2.99	3.12	6.27	**x 7.2**
Rhino to Scene 43	31.06	8.76	39.82	0.46	1.41	0.71	2.58	**x 15.4**

TABLE II
SPEED UP OF OUR APPROACH (CLUSTERING)

Scene	**Number of Points**	**Original DBSCAN**	**Our Appr. oach**	**Speed up**
Scene 3	101,644	18.75	0.46	**x 41.7**
Scene 24	144,130	61.25	0.64	**x 95.7**
Scene 33	112,755	22.53	0.53	**x 42.5**
Scene 42	153,486	68.42	0.70	**x 97.7**
Scene 50	124,464	29.11	0.57	**x 51**

TABLE III
DEFAULT VALUES FOR PARAMETERS

Variable	Default Value	Short description
K_a	5	Number of neighboring vertices in Eq. (6)
K_D	10	Number of neighboring vertices in Eq. (25)
K_s	1000	Value of kernel in Eq. (11)
K_p	200	Number of compared vertices in Eq. (15)
ξ	0.4	Threshold that denotes the dissimilarity

Fig. 6. Rodolà's dataset consisting of 150 synthetic partially-observed scenes.

The first five boxplots present the value of the dissimilarity factor when the *target* model is registered through the broad phase in a scene where a partial representation of the same object also exists (intra-class). On the opposite, the last boxplot presents the values of the dissimilarity factor when the *target* model is

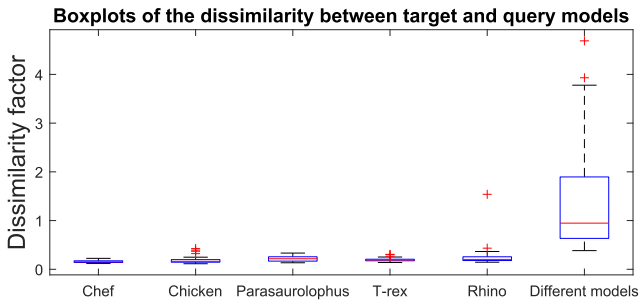

Fig. 7. Range of the dissimilarity factor calculated between each target and query (partial) model when they represent the same object (intra-class) or different objects (inter-class). The first 5 boxplots show intra-class comparisons for every individual object, whereas the last boxplot summarizes the values of all inter-class comparisons.

globally registered in a scene which does not include a partial representation of the same object (inter-class), like in the case of the rhino model as shown in Fig. 2. As we can see, the dissimilarity factor of the first five cases is smaller than 0.5. This means that the global registration provides a correct initial estimate, regardless of the noise, amount of cluttering, and type of object. On the other hand, as we expected, the dissimilarity factor between *query* and *target* models is very high, when they represent different objects.

Fig. 8-[Right] depicts how the occlusion of a model (in percentage %) affects the dissimilarity factor (in case of intra-class matching). In other words, this figure shows that the value of the dissimilarity factor depends on the percentage of the occlusion of the query object. The bigger the occlusion, the higher the possible values of the dissimilarity factor, as observed from the slight shift of the distribution towards higher values. The colors of the heatmap represent the number of the instances per occlusion and dissimilarity factor as presented in the corresponding axes. Fig. 8-[Left] presents a histogram showing the number of instances that in certain range of the dissimilarity factor. The distribution shows that most of the cases have a dissimilarity factor between $[0.1 - 0.3]$. Each bar of the histogram is the total sum of each row of the right figure.

Moreover, we calculated the Precision-Recall (PR) curve which is one of the most common indicators used in the

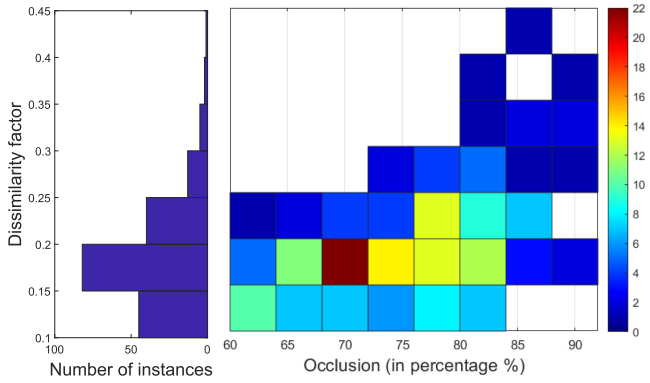

Fig. 8. Effect of occlusion on dissimilarity factor. [Left] Marginal distribution of dissimilarity factor. [Right] Joint distribution of dissimilarity factor and amount of occlusion in the case of intra-class object matching.

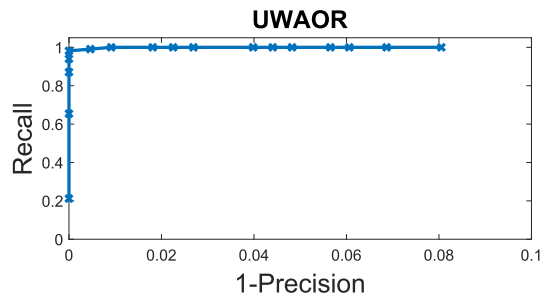

Fig. 9. PR curve under different threshold of the dissimilarity factor.

TABLE IV
RESULTS OF MATCHING THE FIVE TARGET MODELS TO THE 50 PARTIAL SCAN SCENES [UWAOR]

Name of model	Chef	Chicken	Parasaur.	T-rex	Rhino
TP	50/(50)	47/(48)	45/(45)	45/(45)	28/(29)
TN	0/(0)	2/(2)	4/(5)	5/(5)	21/(21)
FP	0	0	1	0	0
FN	0	1	0	0	1

literature for the evaluation of a descriptor or algorithm in retrieval tasks [18]. Precision denotes the number of correct matches to the total number of matches. Recall denotes the number of correct matches to the total number of possible correct matches. In Fig. 9, the PR curve has been created by changing the threshold of the dissimilarity factor (which defines the similarity between two models) in a range of $[0.05 - 1]$ with a step of 0.05. Our method provides a binary decision (if the *query* and *target* models represent the same object or not), based on the value of the predefined threshold. The broad-phase registration only identifies the area of the point cloud scene in which the query model may lie, but the value of the dissimilarity factor is what ultimately defines if there is an actual match. Table IV presents the results of our matching process. The numbers in parenthesis present the correct (existing or non-existing) matches which we aim to identify. For example, the "Chicken" model appears in 48 of the 50 scenes and the method has found 47 true positives (TP) and 2 true negatives (TN) cases. As can be seen, the "Chef" and

TABLE V
STATISTICAL MEASURES FOR THE EVALUATION OF OUR METHOD

	Spec.	Prec.	Recall	Acc.	FPR	F1
Chef	-	1	1	1	-	1
Chicken	1	1	0.979	0.980	0	0.989
Parasaur.	0.800	0.978	1	0.98	0.200	0.989
Tyra	1	1	1	1	0	1
Rhino	1	1	0.965	0.980	0	0.982
Total	**0.969**	**0.995**	**0.991**	**0.988**	**0.030**	**0.993**

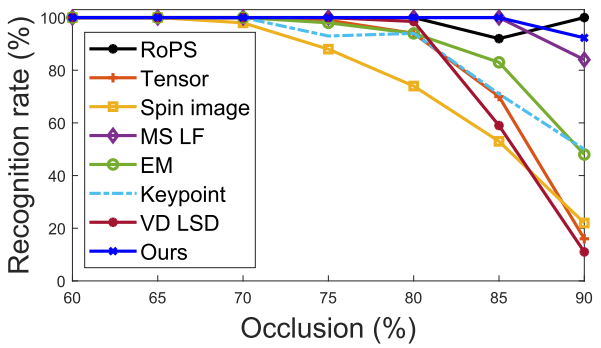

Fig. 10. Recognition rates per different percentage of occlusion, in comparison with other methods, namely tensor [65], spin image [3], keypoint [66], VD-LSD [8], EM based [10], RoPS [11] and MS_LF [2].

the "T-rex" models are correctly matched in all of the 50 and 45 scenes, in which they correspondingly appear. Also, all models except the Parasaurolophus (having one false positive (FP) were correctly identified as missing in all scenes in which they do not appear. Statistical measures of performance are shown in Table V for each model separately as well as averaged across all models. Finally, in Fig. 10, we present the recognition rate, i.e. the fraction of TP over the total number of correct matches, of our method in comparison with other well-known methods of the literature for different occlusion rates. The recognition rate of our method is less than 100% only when the occlusion is higher than $> 85\%$.

2) Evaluation of Matching With Noisy Query Models: In this experimental case, we compare the *target* models with *query* objects that represent a partial scan of the *target* model, under different levels of noise. For the creation of the noisy models, we added, to each *query* model, different levels of Gaussian noise with intensity $\sigma_E = \{0.1, 0.3, 0.5\}$ to each of the original representations [69]. An example of a noisy scene with different levels of Gaussian noise is presented in Fig. 19.

In Table VI, we present the TP and the false negative (FN) matches for each of the different models of the dataset. The TP matches are 100% for the original dataset and under the presence of noise with $\sigma_E = 0.1$. When the level of noise starts to increase (> 0.1), the first false negatives start to appear. Similar conclusions can be observed in Fig. 13. The recognition rate decreases as the level of noise increases, but nevertheless does not drop a lot, as even for a high amount of noise (i.e., $\sigma_E = 0.5$), the mean recognition rate remains high (75%).

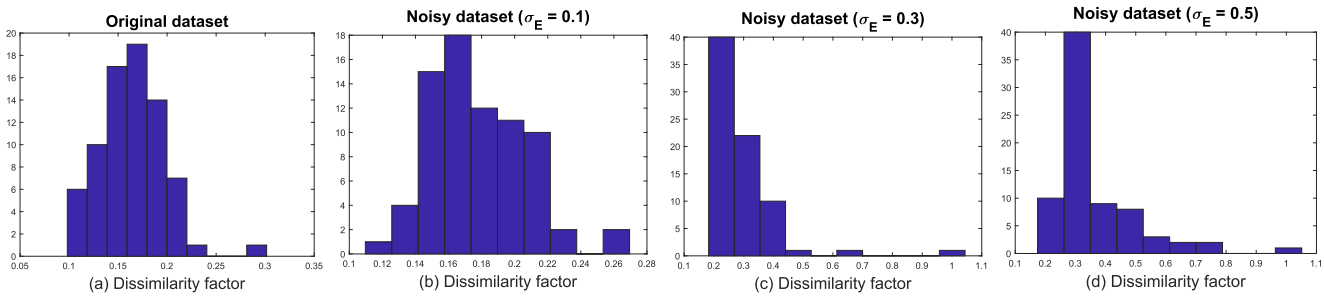

Fig. 11. Effect of noise (σ_E) on dissimilarity factor. (a) original dataset, (b) $\sigma_E = 0.1$, (c) $\sigma_E = 0.3$, (d) $\sigma_E = 0.5$.

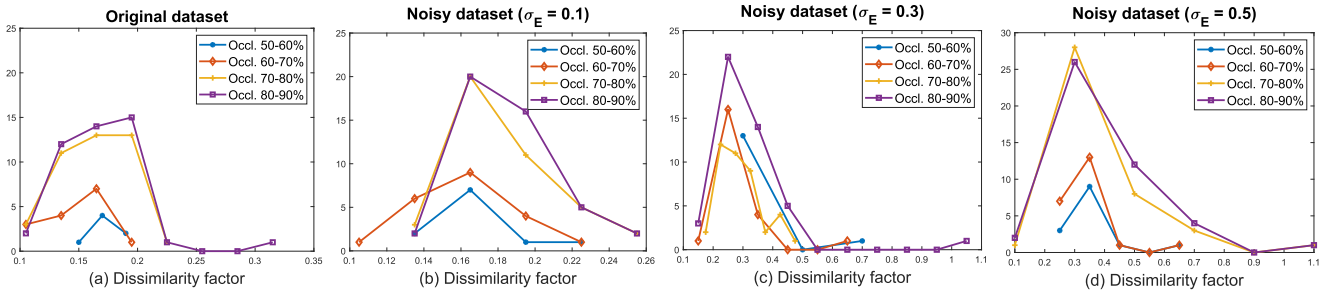

Fig. 12. Distribution of dissimilarity factor between target and query models for different occlusion rate of the query model and noise levels (σ_E) affecting the dataset. (a) original dataset, (b) $\sigma_E = 0.1$, (c) $\sigma_E = 0.3$, and (d) $\sigma_E = 0.5$.

TABLE VI
TRUE POSITIVE AND FALSE NEGATIVE MATCHES USING THE ORIGINAL AND THE NOISY DATASETS [UWA3M]

	Chef	Chicken	Parasaur.	T-rex
Original	TP = 22 FN = 0	TP = 16 FN = 0	TP = 16 FN = 0	TP = 21 FN = 0
Noise $\sigma_E = 0.1$	TP = 22 FN = 0	TP = 16 FN = 0	TP = 16 FN = 0	TP = 21 FN = 0
Noise $\sigma_E = 0.3$	TP = 21 FN = 1	TP = 15 FN = 1	TP = 15 FN = 1	TP = 16 FN = 5
Noise $\sigma_E = 0.5$	TP = 20 FN = 2	TP = 10 FN = 6	TP = 13 FN = 3	TP = 13 FN = 8

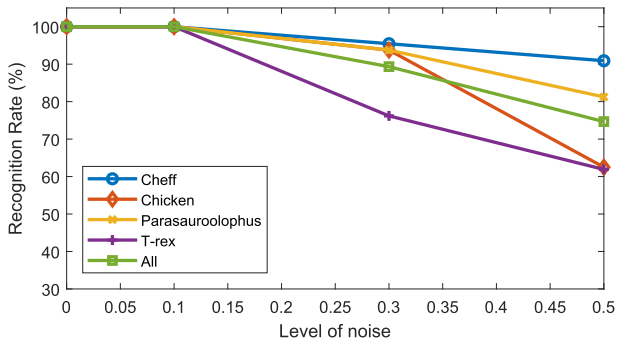

Fig. 13. Recognition rates per different levels of noise for the each model separately and in total.

Fig. 11 shows the histograms representing the number of intra-classes comparisons that have a certain value of dissimilarity (x-axis). As we can observe, in the original dataset and in the dataset that has been affected by $\sigma_E = 0.1$ noise, the dissimilarity factors are less than 0.4. As the amount of noise increases, the dissimilarity factor also increases, affecting the accuracy of object identification. Additionally, in Fig. 12, we present the dissimilarity factor for different percentages of occlusion for the different noisy datasets. As expected, the dissimilarity factor depends on the percentage of the occlusion of the query object, as well as on the level of noise.

3) Evaluation of Segmentation and Registration in Noisy and Low Quality Scenes: Besides the fact that the UWAOR dataset consists of models that have been affected by real noise due to the limitations of the scanner device, we further investigate more challenging situations of noisy and low quality scenes. In Fig. 16, we present how the clustering method works under different levels of Gaussian noise (0.1 - 0.5) and different levels of visual quality (10% - 50% simplification). InFigs. 19, 20, we present some experimental results that show how the performance of the broad-to-narrow registration is affected by different levels of noise and different resolution. More specifically, we applied different levels of Gaussian noise [$\sigma_E = 0.1,\ 0.3,\ 0.5$] to the models, and then we followed the same steps as in the original implementation of our approach. Additionally, in Figs. 19, 20, we present how the low resolution quality of a scene could also affect the performance of the alignment. In this experiment, we downsampled the scene about 10%, 30%, and 50% of the original points (i.e., sim. 10%, sim. 30%, sim. 50%). The experiments show that the alignment of noisy or low-resolution models is deteriorated, as was anticipated. However, the figures and the results show that the performance of the whole proposed pipeline is not significantly affected (in a way to hinder correct matching and registration).

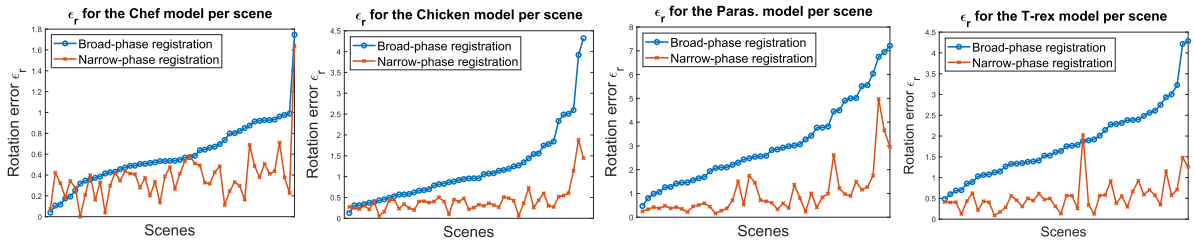

Fig. 14. Rotation error ϵ_r for the broad-phase and the narrow-phase registration for each of the 50 scenes. The scenes are sorted by increasing value of broad-phase registration error.

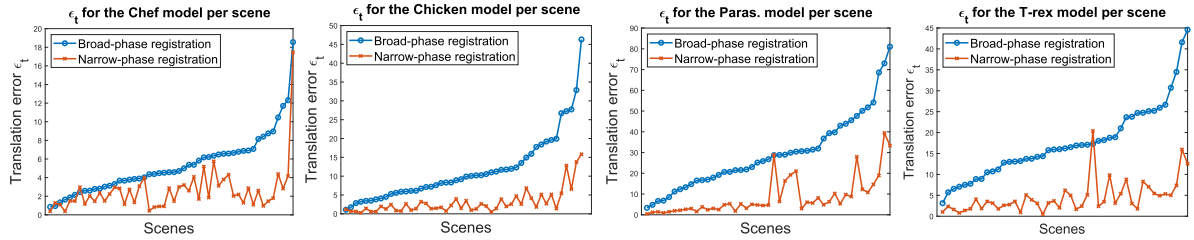

Fig. 15. Translation error ϵ_t for the broad-phase and the narrow-phase registration for each of the 50 scenes. The scenes are sorted by increasing value of broad-phase registration error.

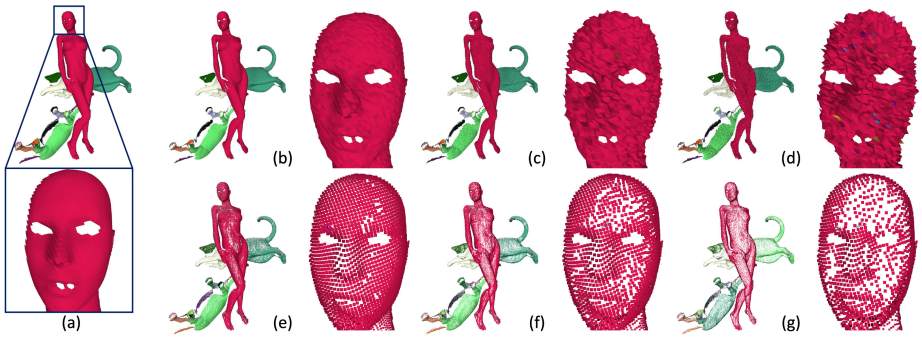

Fig. 16. (a) Original scene clustering and clustering under (b) $\sigma_E = 0.1$, (c) $\sigma_E = 0.3$, (d) $\sigma_E = 0.5$ Gaussian noise, and (e) 10%, (f) 30%, and (g) 50% of total points simplification.

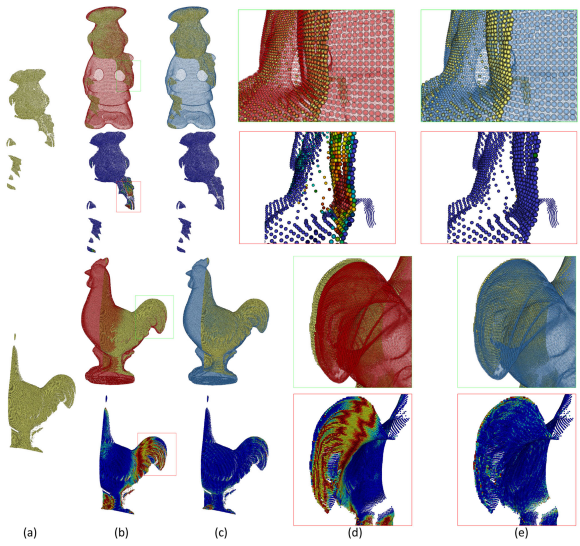

Fig. 17. (a) Partially scanned segments of a scene (b) broad-phase registration and heatmap visualization of mean squared error, (c) narrow-phase registration and heatmap visualization of mean squared error, (d) enlarge details of broad-phase registration, and (e) enlarge details of narrow-phase registration.

4) Evaluation of the Narrow-Phase Registration: Besides the evaluation of the matching process, which results to object identification, we also evaluated the robustness of the registration process that allows the high-quality 3D model to accurately replace the partially scanned model. Even if object identification is always correct, accurate model-to-object registration is not granted.

Figs. 14-15 present the rotation (ϵ_r) and translation (ϵ_t) errors after the broad-phase and the narrow-phase registration of the models in each scene. In the majority of the cases, the narrow-phase registration reduces significantly the error of broad-phase registration. Additionally, the broad-phase registration, used as an initialization step, affects the results of the narrow-phase registration, however the upper bound of the latter remains limited even for inaccurate initializations.

In Fig. 17 we present the results of the broad-phase and the narrow-phase registration for the "Chef" model of the 7^{th} scene and the "Chicken" model of the 49^{th} scene. We also provide enlarged details for an easier comparison between the approaches and heatmaps that visualize the mean squared error between the position of the registered and the original models.

Many methods obtain high registration accuracy when applied to full objects, however, they achieve a much lower accuracy when used with partial objects [70]. Many approaches provide

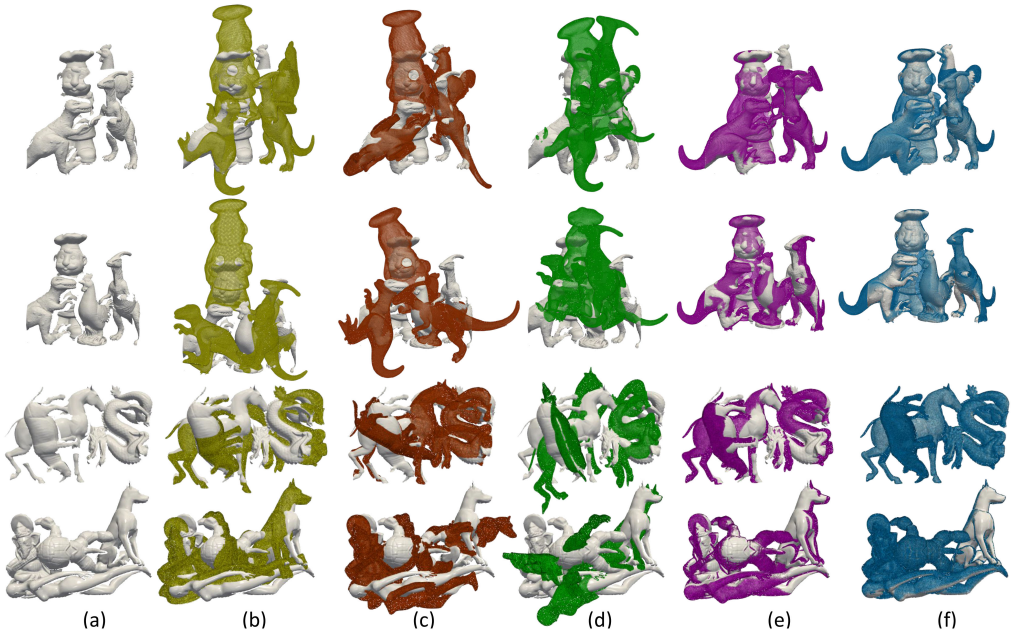

Fig. 18. Visualization of registration results of different methods for 4 random partial scenes. (a) partial scenes consisting of different models in arbitrary positions, registration results using: (b) Robust low-overlap 3-D point cloud registration approach [71], (c) Discriminative Optimization approach [72], (d) Density Adaptive Point Set Registration [73], (e) the Super 4PCS approach [74], and (f) the proposed method.

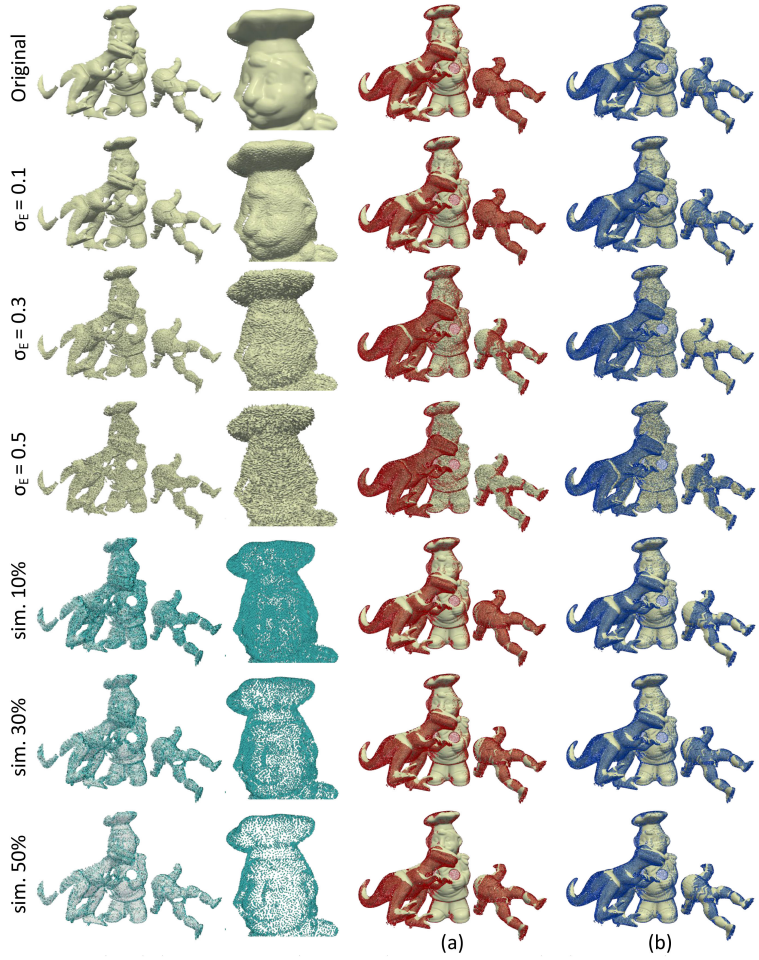

Fig. 19. (a) broad-phase registration and (b) narrow-phase registration, under different conditions.

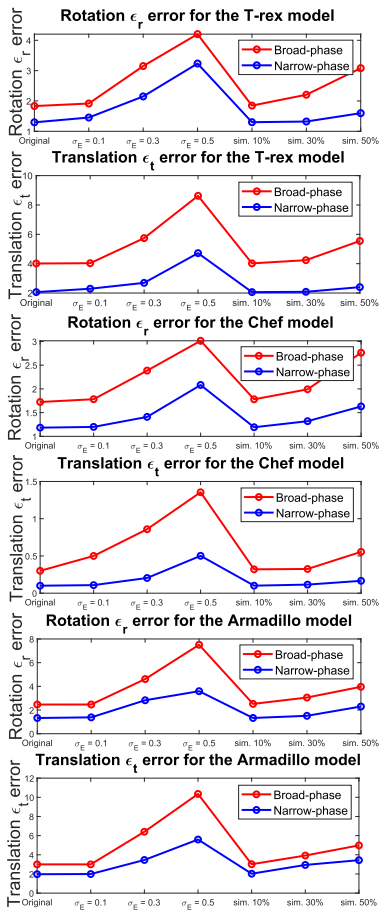

Fig. 20. Rotation ϵ_r and translation ϵ_t error for each method applied to the two datasets (i.e, UWAOR and Rodolà's dataset).

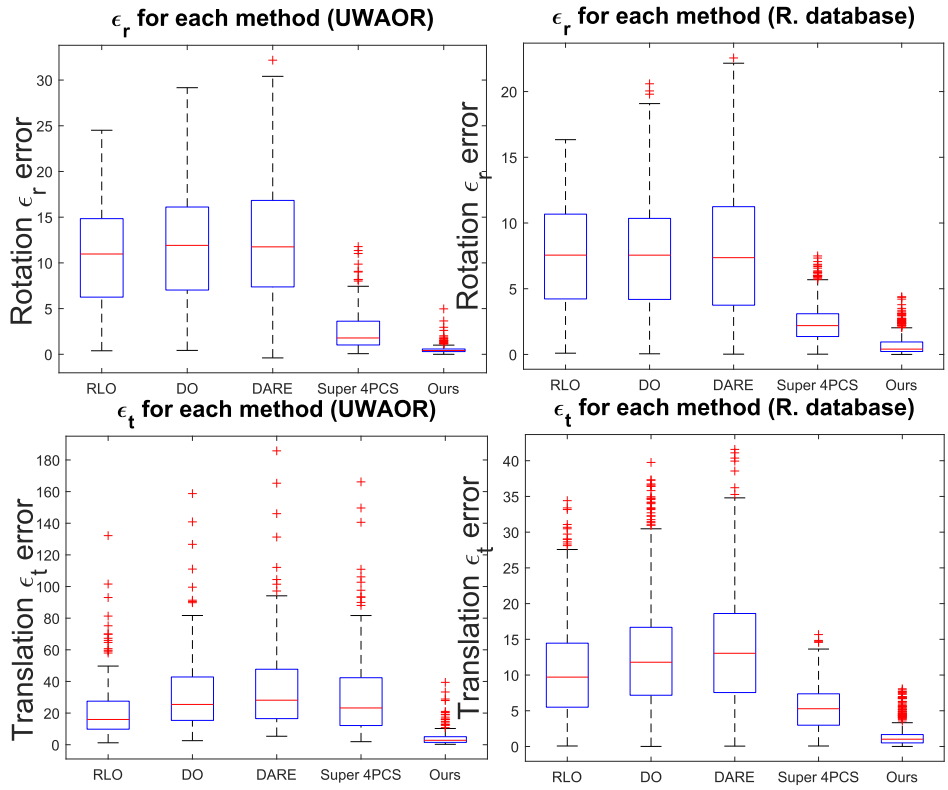

Fig. 21. Rotation ϵ_r and translation ϵ_t error for each method applied to the two datasets (i.e, UWAOR and Rodolà's dataset).

good results only if there is a big overlap between the input models. Other approaches have some constraints, like equally sized models with respect to their number of vertices. In Fig. 18, we present the registration results of different state-of-the-art methods as applied in some partially-scanned scenes of the dataset. More specifically, the used methods for the registration comparisons are: (i) Robust low-overlap 3-D point cloud registration for outlier rejection (RLO) [71], (ii) Discriminative Optimization: Theory and Applications to Point Cloud Registration (DO) [72], (iii) Density Adaptive Point Set Registration (DARE) [73] and (iv) Super 4PCS Fast Global Point cloud Registration via Smart Indexing [74]. To note here that for a fair comparison with all aforementioned approaches, we firstly applied the model-to-object matching results obtained by our method, and then used them for the narrow-phase registration step. The experimental process shows that our robustified weighted ICP-based method provides the most accurate registration for all the models of the partial scenes.

Finally, in Fig. 21, we present the rotation ϵ_r and translation ϵ_t error for each model presented in the 50 scenes of the UWAOR dataset and in the 150 scenes of the Rodolà *et al.* dataset [68], for different competing methods.

V. Conclusion

In this paper, we presented a methodology for the identification and registration of 3D objects in partially scanned and cluttered point clouds. The point clouds might include 3D objects lying in arbitrary positions in multi-object scenes and include noise and outliers. The whole multi-step methodology has been designed to address many challenging subproblems with the aim to reduce uncertainty in matching through dedicated salient point detection and robustification techniques and to optimize the mapping following a broad to narrow registration strategy. The comparison of our method with other state-of-the-art approaches has shown its superiority.

Our future plans include the semantic retrieval of objects in order to complete all partial or low-quality objects of the identified category (e.g., replacement of all partially-observed cars in a cluttered point cloud scene acquired by Lidar sensors, with high-quality car models from a database, regardless of the possible different shapes and forms of the cars). We furthermore aim to investigate the combination of our methodology with non-rigid transformation that will allow to map and replace partially observed deformable objects which appear in different poses (e.g. the moving human body) with the corresponding models of a database.

References

[1] B. Lu and Y. Wang, "Matching algorithm of 3D point clouds based on multiscale features and covariance matrix descriptors," *IEEE Access*, vol. 7, pp. 137 570–137 582, 2019.

[2] M. Lu, Y. Guo, J. Zhang, Y. Ma, and Y. Lei, "Recognizing objects in 3D point clouds with multi-scale local features," *Sensors* (Basel, Switzerland), vol. 14, pp. 24 156-24 173, 2014.

[3] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.

[4] Z. Zhong, J. Zhu, and S. C. H. Hoi, "Fast object retrieval using direct spatial matching," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1391–1397, Aug. 2015.

[5] Y. Sun and M. A. Abidi, "Surface matching by 3D point's fingerprint," in *Proc. 8th IEEE Int. Conf. Comput. Vis. ICCV 2001*, vol. 2, 2001, pp. 263–269 vol.2.

[6] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Comput. Vis. – ECCV*2004, T. Pajdla and J. Matas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 224–237.

[7] S. Malassiotis and M. G. Strintzis, "Snapshots: A novel local surface descriptor and matching algorithm for robust 3D surface alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 7, pp. 1285–1290, Jul. 2007.

[8] B. Taati and M. Greenspan, "Local shape descriptor selection for object recognition in range data," *Comput. Vis. Image Understanding*, vol. 115, no. 5, pp. 681–694, May 2011.

[9] A. Zaharescu, E. Boyer, and R. Horaud, "Keypoints and local descriptors of scalar functions on 2D manifolds," *Int. J. Comput. Vis.*, vol. 100, no. 1, pp. 78–98, Oct. 2012.

[10] P. Bariya, J. Novatnack, G. Schwartz, and K. Nishino, "3D geometric scale variability range images: Features descriptors," *Int. J. Comput. Vis.*, vol. 99, pp. 232–255, 2012.

[11] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, "Rotational projection statistics for 3D local surface description and object recognition," *Int. J. Comput. Vis.*, vol. 105, no. 1, pp. 63–86, 2013.

[12] E. Paquet, M. Rioux, A. Murching, T. Naveen, and A. Tabatabai], "Description of shape information for 2D and 3D objects," *Signal Process.: Image Commun.*, vol. 16, no. 1, pp. 103 - 122, 2000.

[13] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Trans. Graph.*, vol. 21, no. 4, pp. 807–832, Oct. 2002.

[14] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the viewpoint feature histogram," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 2155–2162.

[15] L. Shang, M. Greenspan, "Real-time object recognition in sparse range images using error surface embedding," *Int. J. Comput. Vis.*, CAN, vol. 89, pp. 211–228, 2010. [Online]. Available: https://doi.org/10.1007/s11263-009-0276-3.

[16] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, "3D object recognition in cluttered scenes with local surface features: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2270–2287, Nov. 2014.

[17] S. Salti, F. Tombari, and L. D. Stefano, "Shot: Unique signatures of histograms for surface and texture description," *Comput. Vis. Image Understanding*, vol. 125, pp. 251 - 264, 2014.

[18] A. G. Buch, H. G. Petersen, and N. Krüger, "Local shape feature fusion for improved matching, pose estimation and 3D object recognition," *SpringerPlus*, vol. 5, 2016.

[19] T. Darom and Y. Keller, "Scale-invariant features for 3D mesh models," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2758–2769, May 2012.

[20] Y. Wu, F. He, D. Zhang, and X. Li, "Service-oriented feature-based data exchange for cloud-based design and manufacturing," *IEEE Trans. Services Comput.*, vol. 11, no. 2, pp. 341–353, Mar.-Apr. 2018.

[21] Y. Liang, F. He, and X. Zeng, "3D mesh simplification with feature preservation based on Whale Optimization Algorithm and Differential Evolution," IOS Press, vol. 27, no. 4, pp. 417–435, 2020.

[22] Y. Guo, F. Sohel, M. Bennamoun, J. Wan, and M. Lu, "An accurate and robust range image registration algorithm for 3D object modeling," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1377–1390, Aug. 2014.

[23] H. Liu *et al.*, "Point cloud registration based on MCMC-SA ICP algorithm," *IEEE Access*, vol. 7, pp. 73 637–73 648, 2019.

[24] H. Chen, X. Zhang, S. Du, Z. Wu, and N. Zheng, "A correntropy-based affine iterative closest point algorithm for robust point set registration," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 4, pp. 981–991, Jul. 2019.

[25] H. Kim, H. K. Hong, and B. H. Lee, "Improved iterative closest point algorithm using truncated signed distance function," in *Proc. 18th Int. Conf. Control, Automat. Syst.*, Oct 2018, pp. 1620–1623.

[26] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," in *Proc. 11th Eurographics/ACMSIGGRAPH Symp. Geometry Proc., ser. SGP '13. Goslar, DEU: Eurographics Assoc.*, 2013, pp. 113–123.

[27] P. Mavridis, A. Andreadis, and G. Papaioannou, "Efficient sparse icp," in *Proc. Comput. Aided Geometric Design*, vol. 35–36, pp. 16-26, 2015, *geometric Model. Process.* 2015.

[28] C. Yuan, X. Yu, and Z. Luo, "3D point cloud matching based on principal component analysis and iterative closest point algorithm," in *Proc. Int. Conf. Audio, Lang. Image Process.*, Jul. 2016, pp. 404–408.

[29] W. Tao, X. Hua, K. Yu, X. He, and X. Chen, "An improved point-to-plane registration method for terrestrial laser scanning data," *IEEE Access*, vol. 6, pp. 48 062-48 0 73, 2018.

[30] D. Li, A. Wang, P. Ren, and L. Wu, "An allowance optimal distribution method based on improved iterative closest point algorithm," in *Proc. 10th Int. Conf. Measuring Technol. Mechatronics Automat.*, Feb 2018, pp. 515–518.

[31] Z. Xu, E. Xu, Z. Zhang, and L. Wu, "Multiscale sparse features embedded 4-points congruent sets for global registration of tls point clouds," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 286–290, Feb 2019.

[32] J. Lu, W. Wang, H. Shao, and L. Su, "Point cloud registration algorithm fusing of super 4pcs and icp based on the key points," in *Proc. Chin. Control Conf.*, Jul. 2019, pp. 4439–4444.

[33] J. Sun, R. Zhang, S. Du, L. Zhang, and Y. Liu, "Global adaptive 4-points congruent sets registration for 3D indoor scenes with robust estimation," *IEEE Access*, vol. 8, pp. 7539–7548, 2020.

[34] L. Hruda, J. Dvorˆák, and L. Vášа, "On evaluating consensus in ransac surface registration," *Comput. Graph. Forum*, vol. 38, no. 5, pp. 175–186, 2019.

[35] I. Khan, "Robust sparse and dense nonrigid structure from motion," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 841–850, Apr. 2018.

[36] Y. Keaomanee, P. Youngkong, W. Panpanytep, T. Maneewarn, and S. Vongbunyong, "Kinect v2 edge base registration with ransac-icp," in *Proc. IEEE 4th Int. Conf. Comput. Commun.*, Dec 2018, pp. 1611–1616.

[37] A. Makadia, A. Patterson, and K. Daniilidis, "Fully automatic registration of 3D point clouds," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Jun. 2006, pp. 1297–1304.

[38] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, "The perfect match: 3D point cloud matching with smoothed densities," in *Proc. Int. Conf. Comput. Vis. pattern Recognit.*, 2019, pp. 5540–5549, doi: 10.1109/CVPR.2019.00569.

[39] G. Elbaz, T. Avraham, and A. Fischer, "3D point cloud registration for localization using a deep neural network auto-encoder," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2472–2481.

[40] Z. Zhang *et al.*, "3-d deep feature construction for mobile laser scanning point cloud registration," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 12, pp. 1904–1908, Dec 2019.

[41] G. Truong, S. Z. Gilani, S. M. S. Islam, and D. Suter, "Fast point cloud registration using semantic segmentation," in *Proc. Digit. Image Comput.: Techn. Appl.*, Dec 2019, pp. 1–8.

[42] H. Chen, M. Wei, Y. Sun, X. Xie, and J. Wang, "Multi-patch collaborative point cloud denoising via low-rank recovery with graph constraint," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 11, pp. 3255–3270, Nov. 2020.

[43] W. Hu, X. Gao, G. Cheung, and Z. Guo, "Feature graph learning for 3D point cloud denoising," *IEEE Trans. Signal Process.*, vol. 68, pp. 2841–2856, Mar. 2020.

[44] F. Pistilli, G. Fracastoro, D. Valsesia, and E. Magli, "Learning robust graph-convolutional representations for point cloud denoising," *IEEE J. Sel. Topics Signal Process.*vol. 15, no. 2, pp. 402-414, Feb. 2021.

[45] S. Luo and W. Hu, *Differentiable Manifold Reconstruction for Point Cloud Denoising*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1330–1338. [Online]. Available: https://doi.org/10.1145/3394171.3413727

[46] K. Sarkar, F. Bernard, K. Varanasi, C. Theobalt, and D. Stricker, "Structured low-rank matrix factorization for point-cloud denoising," in *Proc. Int. Conf. 3D Vis.*, 2018, pp. 444–453.

[47] E. Mattei and A. Castrodad, "Point cloud denoising via moving rpca," *Comput. Graph. Forum*, vol. 36, no. 8, pp. 123–137, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13068

[48] S. Fleishman, D. Cohen-Or, and C. T. Silva, "Robust moving least-squares fitting with sharp features," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 544–552, Jul. 2005. [Online]. Available: https://doi.org/10.1145/1073204.1073227

[49] S. Liu, K.-C. Chan, and C. C. L. Wang, "Iterative consolidation of unorganized point clouds," *IEEE Comput. Graph. Appl.*, vol. 32, no. 3, pp. 70–83, May 2012. [Online]. Available: https://doi.org/10.1109/MCG.2011.14

[50] H. Sheung and C. C. L. Wang, "Robust mesh reconstruction from unoriented noisy points," in *Proc. SIAM/ACM Joint Conf. Geometric Phys. Modeling, ser. SPM '09*. New York, NY, USA: Association for Computing Machinery, 2009, pp. 13–24. [Online]. Available: https://doi.org/10.1145/1629255.1629258

[51] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Second Int. Conf. Knowl. Discov. Data Mining, ser. KDD'96*. AAAI Press, 1996, pp. 226–231.

[52] P. N. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces," in *Proc. 4th Annu. ACM-SIAM Symp. Discrete Algorithms, ser. SODA '93*. USA: *Soc. Ind. Appl. Math.*, 1993, pp. 311–321.

[53] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4558–4567.

[54] G. Erus, E. I. Zacharaki, and C. Davatzikos, "Individualized statistical learning from medical image databases: Application to identification of brain lesions," *Med. Image Anal.*, vol. 18, no. 3, pp. 542–554, 2014.

[55] L. Liu, Y. Sheng, G. Zhang, and H. Ugail, "Graph cut based mesh segmentation using feature points and geodesic distance," in *Proc. Int. Conf. Cyberworlds*, Oct 2015, pp. 115–120.

[56] G. Arvanitis, A. S. Lalos, and K. Moustakas, "Saliency mapping for processing 3D meshes in industrial modeling applications," in *Proc. IEEE 17th Int. Conf. Ind. Inform.*, vol. 1, Jul. 2019, pp. 683–686.

[57] Z. Taylor, J. Nieto, and D. Johnson, "Multi-modal sensor calibration using a gradient orientation measure," *J. Field Robot.*, vol. 32, no. 5, pp. 675–695, 2015.

[58] P. Bergström and O. Edlund, "Robust registration of point sets using iteratively reweighted least squares," *Comput. Optim. Appl.*, vol. 58, no. 3, pp. 543–561, Jul 2014.

[59] —, "Robust registration of surfaces using a refined iterative closest point algorithm with a trust region approach," *Numer. Algorithms*, vol. 74, no. 3, pp. 755–779, Mar 2017.

[60] G. Arvanitis, A. S. Lalos, K. Moustakas, and N. Fakotakis, "Outliers removal of highly dense and unorganized point clouds acquired by laser scanners in urban environments," in *Proc. Int. Conf. Cyberworlds*, 2018, pp. 415–418.

[61] G. Arvanitis, A. Spathis-Papadiotis, A. S. Lalos, K. Moustakas, and N. Fakotakis, "Outliers removal and consolidation of dynamic point cloud," in *Proc. 25th IEEE Int. Conf. Image Process.*, 2018, pp. 3888–3892.

[62] G. A. Arvanitis S. Lalos, K. Moustakas, and N. Fakotakis, "Real-Time Removing of Outliers and Noise in 3D Point Clouds Applied in Robotic Applications," in Interactive Collaborative Robotics, A. G. Ronzhin Rigoll, and R. Meshcheryakov, Eds. Cham: Springer International Publishing, 2017, pp. 11–19.

[63] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, p. 11, 2011.

[64] Z. Lin, M. Chen, and Y. Ma, "The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices," CoRR, vol. abs/1009.5055, 2009.

[65] A. S. Mian, M. Bennamoun, and R. Owens, "Three-dimensional model-based object recognition and segmentation in cluttered scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1584–1601, Oct 2006.

[66] A. Mian, M. Bennamoun, and R. Owens, "On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes," *Int. J. Comput. Vis.*, vol. 89, no. 2, pp. 348–361, Sep 2010.

[67] A. S. Mian, M. Bennamoun, and R. A. Owens, "A novel representation and feature matching algorithm for automatic pairwise registration of range images," *Int. J. Comput. Vis.*, vol. 66, no. 1, pp. 19–40, Jan. 2006.

[68] E. Rodolá, A. Albarelli, F. Bergamasco, and A. Torsello, "A scale independent selection process for 3D object recognition in cluttered scenes," *Int. J. Comput. Vis.*, vol. 102, no. 1, pp. 129–145, 2013.

[69] G. Arvanitis, A. S. Lalos, K. Moustakas, and N. Fakotakis, "Feature preserving mesh denoising based on graph spectral processing," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 3, pp. 1513–1527, Mar. 2019.

[70] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 223–232, 2003.

[71] J. Stechschulte, N. Ahmed, and C. Heckman, "Robust low-overlap 3-d point cloud registration for outlier rejection," in *2019 Int. Conf. Robot. Automat.*, 2019, pp. 7143–7149.

[72] J. Vongkulbhisal, F. De la Torre, and J. P. Costeira, "Discriminative optimization: Theory and applications to point cloud registration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3975–3983.

[73] F. J. Lawin, M. Danelljan, F. S. Khan, P. Forssén, and M. Felsberg, "Density adaptive point set registration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3829–3837.

[74] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4pcs fast global pointcloud registration via smart indexing," *Comput. Graph. Forum*, vol. 33, no. 5, pp. 205–215, 2014.