A benchmark of simulated range images for partial shape retrieval

**Ivan Sipiran · Rafael Meruane · Benjamin Bustos ·
Tobias Schreck · Bo Li · Yijuan Lu · Henry Johan**

Abstract In this paper, we address the evaluation of algorithms for partial shape retrieval using a large-scale simulated benchmark of partial views which are used as queries. Since the scanning of real objects is a time-consuming task, we create a simulation that generates a set of views from a target model and at different levels of complexity (amount of missing data). In total, our benchmark contains 7,200 partial views. Furthermore, we propose the use of weighted effectiveness measures based on the complexity of a query. With these characteristics, we aim at jointly evaluating the effectiveness, efficiency and robustness of existing algorithms. As a result of our evaluation, we found that a combination of methods provides the best effectiveness, mainly due to the complementary information that they deliver. The obtained results open new questions regarding the difficulty of the partial shape retrieval problem. As a consequence, potential future directions are also identified.

1 Introduction

Three-dimensional data are attracting the attention of many research fields due to the potential applications in real scenarios. On one side, the availability of large repositories of shapes, such as Google 3D Warehouse and TurboSquid, open up new possibilities for the exploration of effective and efficient content-based search engines that provide support to further high-level applications in modeling, engineering, and so on. On the other side, the availability of consumer-level 3D scanning devices has promoted the massive use of 3D data for scene understanding and vision-related topics. Nevertheless, a non-trivial question remains open about how to take advantage of these two resources jointly. One possible scenario is the content-based retrieval when the query is a scanned object. The ability of querying a 3D shape repository using a scanned object can support high-level tasks such as recognition, modeling with examples, engineering processes, just to name a few.

In the shape retrieval community, this problem is called partial shape retrieval (or whole-from-part retrieval) and it is still an open and challenging problem. The problem can be stated as: given a partial view of an object as query, retrieve all the 3D models from a repository which are partially similar. By partial similarity, we mean objects that have some part which is similar to the query. In special for scanned objects,

I. Sipiran (✉) · T. Schreck
Department of Computer and Information Sciences,
University of Konstanz, Konstanz, Germany
e-mail: ivan.sipiran@gmail.com;
sipiran@dbvis.inf.uni-konstanz.de

T. Schreck
e-mail: tobias.schreck@uni-konstanz.de

R. Meruane · B. Bustos
Department of Computer Science,
University of Chile, Santiago, Chile
e-mail: rmeruane@dcc.uchile.cl

B. Bustos
e-mail: bebustos@dcc.uchile.cl

B. Li · Y. Lu
Department of Computer Science,
Texas State University, San Marcos, USA
e-mail: b_l58@txstate.edu

Y. Lu
e-mail: lu@txstate.edu

H. Johan
Fraunhofer IDM@NTU, Singapore, Singapore
e-mail: hjohan@fraunhofer.sg

Table 1 Characteristics of partial shape retrieval datasets

Dataset	# Target	# Queries
SHREC 2007 [26]	400	30
SHREC 2009 [7]	720	20
SHREC 2010 [8]	800	120
Our benchmark	360	7,200

we would like to retrieve objects such that some part of its surface fits the geometry of the view.

A problem that requires attention is the lack of a standard framework to evaluate algorithms in partial shape retrieval. In this paper, we aim at evaluating algorithms for partial shape retrieval using a large set of queries composed of views extracted from a 3D dataset. The manual creation of 3D view data for benchmarking is a time-consuming and expensive approach which is expected to be not scalable for the creation of large benchmarks. Therefore, our main idea is to simulate a large number of partial views from an existing 3D object benchmark by generating point clouds from a number of views of a model. Furthermore, we promote the evaluation of the following criteria: effectiveness, efficiency and robustness. To evaluate effectiveness, we rely on common methodologies borrowed from the information retrieval community. To evaluate the efficiency, we measure the query time. Finally, to evaluate the robustness, we propose the use of weighted effectiveness measures which can provide a better understanding about the robustness of the algorithms to the amount of missing data. To do so, each query has an associated factor derived from the amount of missing data it represents. In this way, an algorithm is more robust if challenging queries obtain a good effectiveness.

Previous datasets have been presented so far in past editions of shape retrieval contest (SHREC) [7,8,26] trying to evaluate partial retrieval algorithms. Nevertheless, the query sets are rather small, with dozens of query views provided. In contrast, in our benchmark, a query set composed of 7,200 3D views, obtained from 360 target models is provided. Compared to standard datasets in the 3D retrieval community, this query set can be considered a large-scale evaluation benchmark. Table 1 shows some characteristics of previous benchmarks compared to our proposed dataset. It is also worth noting that previous efforts have been made to build benchmarks with scanned objects in the context of object recognition [5,11]. However, in those cases, the 3D object of the scanned object is often not available and therefore it is not possible to use algorithms based on the geometry of the target model. In contrast, our benchmark is designed to evaluate how well the algorithms assess the similarity between a partial view and a 3D object, which is probably stored in a shape repository.

This paper is an extension of a SHREC track [24], where we introduced the dataset and provided a preliminary evaluation of two methods. Compared to the previous paper, we now provide a comprehensive evaluation of five techniques. In addition, we present a detailed description of our method for generating the views to make easy the reproduction of our methodology in other contexts if required. Furthermore, additionally to the evaluation of each method, we show that the combination of methods exhibits better effectiveness due to the complementary characteristics in each method.

The contributions of our paper can be summarized as follows:

- We create a large set of partial views by simulating the acquisition process of 3D scanners.
- We make a comprehensive evaluation of five algorithms for partial shape retrieval. Basically, the five algorithms fall into three common approaches for partial retrieval: view-based, partition-based and bag of features.
- We define weighted effectiveness measures that incorporate the knowledge about partiality of a query. These measures are useful to evaluate the robustness of the algorithms against missing data.

The paper is organized as follows. Section 2 presents the dataset and how it was built. Section 3 introduces the evaluation methodology. Section 4 is devoted to describe the five evaluated approaches which were submitted for evaluation. Section 5 evaluates and discusses the obtained results. Finally, Sect. 6 draws our conclusions and lists several promising directions as the future work.

2 Construction of the benchmark based on simulated range views

The dataset[1] is divided into two parts: the target set and the query set. The target set is composed of a subset of the SHREC 2009 generic shape retrieval dataset [7]. This dataset provides a uniform distribution of class sizes, thereby avoiding class bias during the evaluation. We chose 360 shapes organized into 20 classes of 18 objects per class. Figure 1 shows one example for each class in the target set. On the other hand, to obtain the query set, we simulate the process of range scan acquisition based on the target set to obtain a set of partial views. The detailed steps to obtain the query set are listed below.

---

[1] The dataset and the evaluation software is available in http://dataset.dcc.uchile.cl.

Fig. 1 Classes in the target set. The classes are listed in a row-based manner from *left* to *right*: bird, fish, insect, biped, quadruped, bottle, cup, mug, floor lamp, desk lamp, cellphone, deskphone, bed, chair, wheelchair, sofa, biplane, monoplane, car and bicycle

– A target shape is enclosed in a regular icosahedron. Beforehand, the shape is translated to the origin of the coordinate system and scaled to fit into a unit cube.
– Each triangular face of the icosahedron will be used as a projection plane. The intersecting points between the object and the rays leaving the projection plane generate a 3D point set. For more details about the point cloud construction, see Algorithm 1.
– A 3D mesh is reconstructed from the obtained point set using the Point Cloud Library [21] using the Greedy Projection Triangulation method. In brief, this method works incrementally for each point. It selects a number of neighbors in the sphere of radius $r = \mu \times d_0$ centered at the analyzed point (d_0 is the distance to the nearest neighbor of the analyzed point). The neighborhood is projected into a tangential plane and a visibility analysis is done. Finally, the neighborhood is connected forming triangles with angle constraint. We set the nearest neighbor distance multiplier μ to be 2.5 and the number of nearest neighbors is set to 20. In addition, we applied a simple hole filling algorithm to discard small holes. More specifically, our algorithm creates a new face when three adjacent faces share a triangle hole.

This simulation process represents a simplified framework of a 3D data acquisition pipeline, including a moderate degree of postprocessing (mesh generation) which is often included in current 3D acquisition software. While more complex

---

Algorithm 1

Require: Shape X (list of triangles)
Require: Icosahedron face $T = (P_1, P_2, P_3)$
Require: Raster resolution r of the projection plane
Ensure: Point cloud C
1: Create a grid of points in the XY plane which will be the raster
2: $spacing = 2/r$
3: $xRaster = [spacing/2 - 1 : spacing : 1 - spacing/2]$
4: $yRaster = [spacing/2 - 1 : spacing : 1 - spacing/2]$
5: Compute the barycenter $B = (P_1 + P_2 + P_3)/3$
6: $\overrightarrow{B_n} = \overrightarrow{OB}/\|\overrightarrow{OB}\|$ (O is the point [0 0 0])
7: $\overrightarrow{u} = \overrightarrow{P_2 P_3}/\|\overrightarrow{P_2 P_3}\|$
8: $\overrightarrow{v} = \overrightarrow{BP_1}/\|\overrightarrow{BP_1}\|$
9: $\overrightarrow{w} = \overrightarrow{u} \times \overrightarrow{v}$
10: Create a local reference frame with vectors $\overrightarrow{u}$, $\overrightarrow{v}$ and $\overrightarrow{w}$. Let R be the rotation matrix defined as
11:

$$R = \begin{pmatrix} \overrightarrow{u}_x & \overrightarrow{v}_x & \overrightarrow{w}_x \\ \overrightarrow{u}_y & \overrightarrow{v}_y & \overrightarrow{w}_y \\ \overrightarrow{u}_z & \overrightarrow{v}_z & \overrightarrow{w}_z \end{pmatrix}$$

12: **if** $\overrightarrow{B_n} \cdot \overrightarrow{w} < 0$ **then**
13:     $\overrightarrow{u} = -\overrightarrow{u}$
14:     $\overrightarrow{w} = -\overrightarrow{w}$
15: **end if**
16: **for** each x_i in $xRaster$ **do**
17:     **for** each y_i in $yRaster$ **do**
18:   Compute the projection point p

$$p = R \times [x_i \ y_i \ 0]^T + B$$

19:         Create a ray Q from p parallel to $\overrightarrow{w}$
20:         Search for intersections between Q and X
21:         **if** number of intersections > 0 **then**
22:             Pick the closest intersection point p_c to p
23:             Insert p_c in the point cloud C
24:         **end if**
25:     **end for**
26: **end for**

---

modifications (in particular adding noises) could be considered, we believe this framework is a valid first step. Figure 2 shows the stages of our simulated acquisition. In total, our method generates 20 partial views for each target mesh, so the complete query set contains 7,200 queries.

At this point, we want to make an observation about the generated partial views. The size and quality of the partial views depend on both the object and the point of view. So it is possible that some views contain less information than others. Therefore, there is an important factor that we need to take into account: how partial is a view with respect to the original mesh? To deal with this aspect, we attach a partiality factor to each partial view which can be considered as a measure of difficulty. The partiality is defined as the surface area ratio between the partial view and the original shape. This factor will be used to weight the retrieval performance as we will show in Sect. 3.
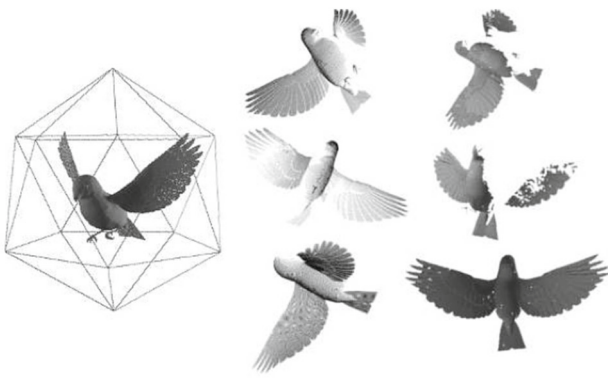
Fig. 2 Process to obtain the dataset. *Left* a shape is enclosed in a regular icosahedron. *Middle* a set of point clouds is obtained by projecting the shape onto each face of the icosahedron. *Right* Meshes are then reconstructed from the point clouds, after a hole filling method has been applied

3 Methodology

The input of the evaluation is a distance matrix of $7,200 \times 360$ where each entry stores the distance between a query view and a target model. Note that each query object was used for measuring the individual performance and then final measures were obtained by averaging over the complete set of queries. For evaluation, we used measures based on precision and recall to analyze the effectiveness of the algorithms. For a given query, precision is the ratio of retrieved relevant objects with respect to the complete list of retrieved objects. Likewise, recall is the ratio of retrieved relevant objects with respect to the complete list of relevant objects.

We use four standard measures commonly used by the information retrieval community:

– *Mean average precision (MAP)* Given a query, its average precision is the average of all precision values computed on all relevant objects in the retrieved list. Given several queries, the mean average precision is the mean of average precision of each query.
– *Nearest neighbor (NN)* Given a query, it is the precision on the first retrieved object in the ranked list. For a set of queries, NN is the average of nearest neighbor measures.
– *First tier (FT)* Given a query, it is the precision when C objects have been retrieved, where C is the number of relevant objects in the 3D dataset. For a set of queries, FT is the average of first tier measures.
– *Second tier (ST)* Given a query, it is the precision when $2 \times C$ objects have been retrieved, where C is the number of relevant objects in the 3D dataset. For a set of queries, ST is the average of second tier measures.

Furthermore, we use a rank-based measure to evaluate the effectiveness of retrieving the exact target object corresponding to a given partial view query,

– *Mean query rank (MQR)* Given a query, the query rank is the position (in the ranked list) of the object in the dataset which generated that query (partial view). Given several queries, the mean query rank is the mean of query ranks for each query.

The aforementioned measures do not consider the relative complexity of each query. In this case, the dataset provides the information about partiality which is a good indicator of complexity. Therefore, we use a weighted version of each effectiveness measure as follows. For the precision-based measures (MAP, NN, FT and ST) and a set of queries Q with their partiality information, the weighted version is

$$measure_w = \frac{\sum_{i=1}^{|Q|}(1 - partiality(i)) \times measure}{\sum_{i=1}^{|Q|}(1 - partiality(i))} \quad (1)$$

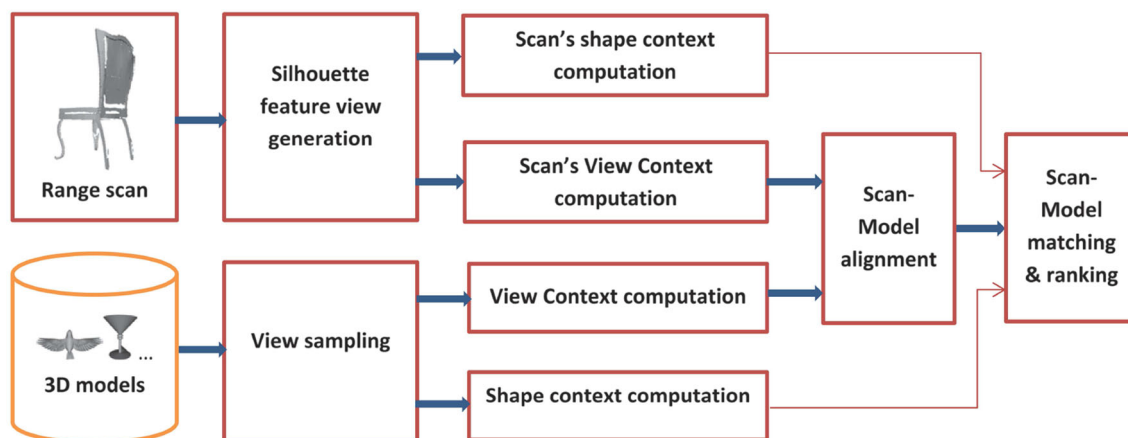For the rank-based measure (MQR), we use the following weighted counterpart

$$measure_w = \frac{\sum_{i=1}^{|Q|} partiality(i) \times measure}{\sum_{i=1}^{|Q|} partiality(i)}. \quad (2)$$

Note that the weights contribute to enhance the measures when $partiality(i)$ gets smaller. For the precision-based measures, a small $partiality(i)$ improves the performance. Similarly, for the rank-based measure, a small $partiality(i)$ contributes to decrease the rank.
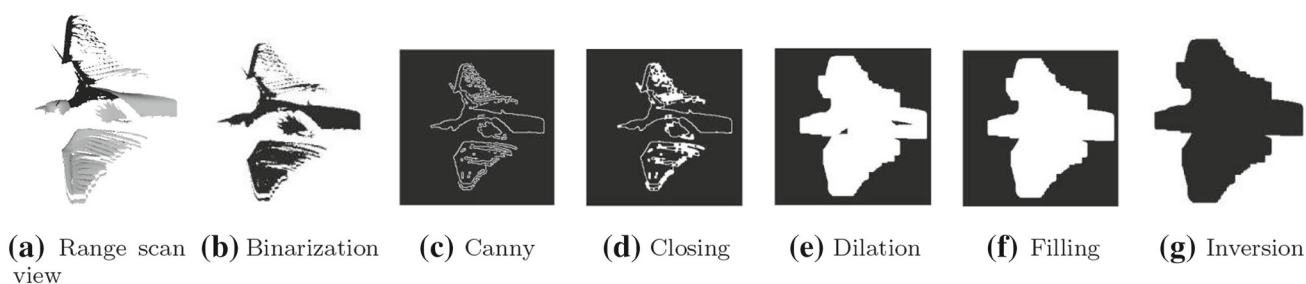
4 Methods

We evaluate five methods for partial shape retrieval. Following is a list of contributions and the authors.

– Range scan-based 3D model retrieval by incorporating 2D–3D alignment by Li et al. [14,16]. This method is presented in Sect. 4.1 (for abbreviation, we refer this method as SBR-2D–3D).
– Range scan-based 3D model retrieval by viewpoint entropy-based adaptive view clustering by Li et al. [15]. This method is presented in Sect. 4.2 (for abbreviation, we refer this method as SBR-VC).
– Partial shape retrieval using data-aware partitioning by Sipiran et al. [23]. This method is presented in Sect. 4.3 (for abbreviation, we refer this method as data-aware).
– Partial shape retrieval using Bag of Features by Sipiran et al. This method is presented in Sect. 4.4 (for abbreviation, we refer this method as BoF).
– Partial shape retrieval with spin images and signature quadratic form distance by Sipiran and Bustos. This

Fig. 3 Flowchart of the range scan-based 3D model retrieval algorithm

(a) Range scan view **(b)** Binarization **(c)** Canny **(d)** Closing **(e)** Dilation **(f)** Filling **(g)** Inversion

Fig. 4 Silhouette feature view generation from a range scan view image

method is presented in Sect. 4.5 (for abbreviation, we refer this method as SQFD).

The evaluation of the methods is performed in two parts. First, each method is evaluated with different values for its involved parameters. These evaluations are presented right after the description of the method. Finally, the best configuration of each method is compared in Sect. 5.

4.1 Range scan-based 3D model retrieval by incorporating 2D–3D alignment

The retrieval algorithm is a modified version of the sketch-based 3D model retrieval algorithm proposed in [14]. The main steps are described in Fig. 3. It comprises precomputation and online retrieval which contains two successive steps: scan-model alignment and scan-model matching. In detail, it first precomputes the view context [13] and relative shape context features (100 sample points) of a set of (e.g., 81 in the algorithm) densely sampled views for each model in the 3D dataset. For the query scan, we first generate its silhouette feature view and then similarly compute its view context and relative shape context features. Based on the view context of the silhouette feature view and the sample views of a 3D model, we perform a scan-model alignment by shortlisting several (8 or 16 in this case) candidate views of the model

to correspond with the silhouette feature view and finally perform scan-model matching based on the shape context matching between the silhouette feature view of the query scan and the candidate sample views of the 3D model.

To extract the relative shape context features and compute the view context feature for a range scan query, it is required to first generate its silhouette feature view. This is also the main difference between the modified retrieval algorithm for range scan queries and the original algorithm for sketch queries in [14,16]. The details of the silhouette feature view generation for the range scan query are as follows. The operations applied to a query to obtain its silhouette feature view are summarized in Fig. 4.

We summarize the range scan-based 3D model retrieval algorithm based on a similar 2D–3D alignment process [14] into the following six steps. Given a simulated query scan and the target 3D model dataset, we perform the retrieval as below.

(1) *Feature views generation* We generate both silhouette and outline feature views for the scan and each 3D model.

(2) *2D feature distance computation* Two different feature distances are utilized hierarchically. To efficiently compute the view context shape descriptor, we use an integrated image descriptor ZFEC. It is composed of four components: Zernike

Table 2 Performance measures of SBR-2D–3D (without partiality)

Run	NN	FT	ST	MAP	MQR
(#CV = 16)	0.3535	0.2290	0.1808	0.2455	62.7326
(#CV = 8)	0.3456	0.2205	0.1736	0.2350	66.1258
(#CV = 4)	0.3444	0.2117	0.1675	0.2248	71.9232

Table 3 Performance measures of SBR-2D–3D (with partiality)

Run	NN	FT	ST	MAP	MQR
(#CV = 16)	0.3504	0.2279	0.1803	0.2447	56.7151
(#CV = 8)	0.3417	0.2194	0.1731	0.2342	60.0114
(#CV = 4)	0.3399	0.2106	0.1670	0.2240	66.4191

moments feature Z of the silhouette view, Fourier descriptors F of the outline view, eccentricity feature E and circularity feature C of the outline view. To more accurately calculate the distance between the scan and each of the shortlisted candidate view of a 3D model, we perform the relative shape context matching [2] between them.

(3) *Scan's view context computation* By computing the ZFEC feature distances between the range scan r and all the base views of each 3D model, we obtain a series of distances $D^r = \langle d_1, d_2, \ldots, d_m \rangle$, which represent the scan's view context, where d_i ($1 \leq i \leq m$) is the distance between the scan and the ith base view of the model.

(4) *Scan-model alignment* We align a 3D model with the scan by shortlisting a certain percentage (e.g., 20/10/5 %, that is 16/8/4 sample views for this retrieval task; default value is 20 % or 16 views) of candidate views with top view context similarities as the scan, in terms of correlation similarity $S_i = \frac{D_i^s \cdot D^r}{\|D_i^s\| \|D^r\|}$. D_i^s and D^r are the view contexts of the ith sample view V_i^s of the 3D model and the scan, respectively.

(5) *Parallel scan-model distance computation* We perform a parallel relative shape context matching between the outline feature view of the scan and every candidate outline feature view and the minimum relative shape context matching distance is deemed as the scan-model distance. Besides parallelization, we also optimize the relative shape context computation and these two improvements work help a lot to accelerate the retrieval process, compared to the initial algorithm proposed in [14].

(6) *Ranking and output* All the scan-model distances are sorted ascendingly and the models are finally listed accordingly as the final retrieval result.

The three runs SBR-2D–3D (#CV = 16), SBR-2D–3D (#CV = 8), SBR-2D-3D (#CV = 4) are three variations with different number of candidate views. Table 2 shows the results of SBR-2D–3D method using the unweighted mea-

sures. Table 3 shows the results of SBR-2D–3D method using the weighted measures.

As can be seen in Table 2, if the number of candidate views is reduced half from the default value of 16, that is, 8 views, the average NN, FT, ST, MAP and MQR scores decrease only 2.23, 3.71, 3.98, 4.28 and 5.41 %, respectively. Even when we reduce it further to be only $\frac{1}{4}$ of the default value, that is, only 4 candidate views, the corresponding scores drop only 2.57, 7.55, 7.36, 8.43 and 14.65 %. We will reach similar conclusions if we measure the performance decreases based on their weighted versions. This again demonstrates the good efficiency property of SBR-2D–3D w.r.t the number of candidate views when it is applied to range scan-based 3D model retrieval.

Another important criterion to take into account is the robustness to partiality. When the weighted measures are used, all the precision-based measures drop consistently (see Tables 2, 3). This means that there is a high probability that challenging queries are obtaining a low precision. We believe that highly partial queries are difficult to represent with their view context as they convey poor silhouette information. Also, the selection of candidate views based on alignment could be affected since the processing is more global and partial matching is not being considered.

4.2 Range scan-based 3D model retrieval by viewpoint entropy-based adaptive view clustering

This method is motivated by the finding that usually different numbers of sample views are needed to represent different 3D models because they differ in visual complexities. A 3D model visual complexity metric is proposed first by utilizing the viewpoint entropy distribution of 81 sample views of a model. Then, it is used to adaptively decide the number of the representative views of the 3D model to perform a Fuzzy C-means view clustering on its 81 sample views. Finally, during online retrieval it performs a more accurate and parallel relative shape context matching [2] (same implementation as that in Sect. 4.1) between a query sketch and the representative views for each target model. The modified range scan-based 3D model retrieval algorithm contains two stages: precomputation and online retrieval, as illustrated in Fig. 5.

The silhouette and outline feature view generation processes are the same as those in Sect. 4.1. Viewpoint entropy-based adaptive view clustering is a most important part of the SBR-VC retrieval algorithm. It is composed of the following three steps.

(1) *Viewpoint entropy distribution* We sample a set of (e.g., 81 in the algorithm) viewpoints for each 3D model. Then, we compute entropy for each viewpoint based on the method in [25]. Fig. 6 demonstrates the viewpoint entropy distribu-
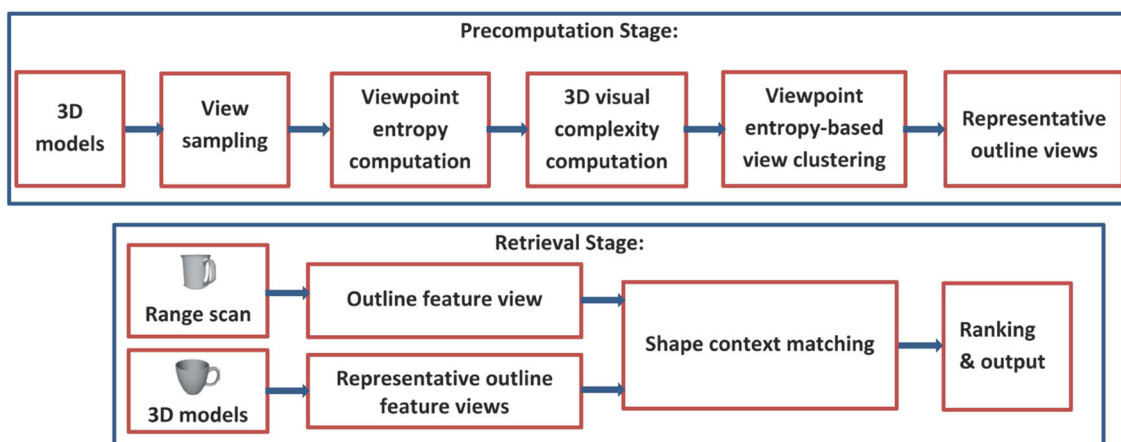
Fig. 5 An overview of the SBR-VC algorithm. The first row is for precomputation while the second row is for retrieval stage

Fig. 6 Viewpoint entropy distribution examples: *first row* shows the models (in the original poses); *second row* demonstrates the viewpoint entropy distributions of the models seen from the original poses. Entropy values are mapped as *colors* on the surface of the spheres based on HSV color model and smooth shading. *Red* small entropy; *green* mid-size entropy; *blue* large entropy

(a) bird   **(b)** Max Planck   **(c)** armadillo

(d) bird   **(e)** Max Planck   **(f)** armadillo
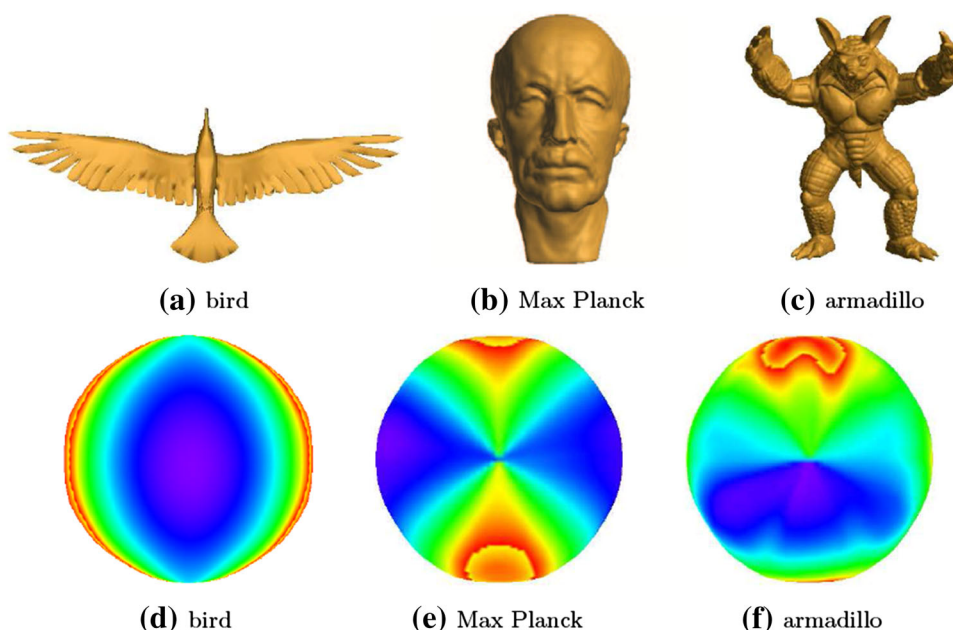
tions of three models based on L_3 for view sampling. As can be seen, there is high correlation between the geometric complexity of a 3D model and the complexity of its entropy distribution pattern. For example, the two complex models Max Planck and armadillo have shown more complicated entropy distribution patterns than the relatively simpler model bird.

(2) *Viewpoint entropy-based 3D visual complexity* The original visual complexity metric proposed in [15] is based on a class-level entropy distribution analysis on a 3D dataset. Since the class information of the target 3D dataset is unavailable, we modified its definition by computing the visual complexity per model. We first compute the mean and standard deviation entropy values E_m and E_s among all the sample views of each 3D model. 3D visual complexity C is defined as $C = \sqrt{\frac{\widehat{E_s}^2 + \widehat{E_m}^2}{2}}$, where $\widehat{E_s}$ and $\widehat{E_m}$ are the normalized E_s and E_m by their respective maximum and minimum over all

the models. $C \in [0, 1]$. The metric reasonably reflects the semantic distances among different types of models.

(3) *Viewpoint entropy-based adaptive views clustering* Based on the 3D visual complexity value C of a 3D model, the number of its representative outline feature views N_c is adaptively assigned: $N_c = \lceil \alpha \cdot C \cdot N_0 \rceil$, where α is a constant and N_0 is the total number of sample views and it is set to 81 in the algorithm. To speed up the retrieval process, α is set to $\frac{1}{2}$ or $\frac{1}{3}$, which corresponds to averagely 14.6 or 9.6 representative views over all the models in the dataset. Finally, to obtain the representative views a Fuzzy C-Means view clustering is performed based on the viewpoint entropy values and viewpoint locations of its 81 sample views.

The two runs, SBR-VC ($\alpha = \frac{1}{2}$) and SBR-VC ($\alpha = \frac{1}{3}$), are two variations of the above SBR-VC algorithm by setting

Table 4 Performance measures of SBR-VC (without partiality)

Run	NN	FT	ST	MAP	MQR
$(\alpha = 1/2)$	0.3218	0.2065	0.1638	0.2199	75.3940
$(\alpha = 1/3)$	0.3025	0.1994	0.1585	0.2117	80.8662

Table 5 Performance measures of SBR-VC (with partiality)

Run	NN	FT	ST	MAP	MQR
$(\alpha = 1/2)$	0.3209	0.2066	0.1639	0.2202	70.9168
$(\alpha = 1/3)$	0.3010	0.1996	0.1587	0.2121	75.6762

different α values, while the number of sample points for the contour(s) of each sketch is set to 100. For more details about the SBR-VC retrieval algorithm, please refer to [15].

Table 4 shows the results of SBR-VC method using the unweighted measures. Table 5 shows the results of SBR-VC method using the weighted measures. As can be seen, SBR-VC achieves a comparable performance as SBR-2D–3D while their main difference is in the view selection strategy. Therefore, the view clustering approach used in SBR-VC achieves a similar effect as that of the scan-model alignment process in SBR-2D–3D. However, SBR-VC needs less computation, saving space and loading memory for the view selection process since it only precomputes, saves and loads the relative shape context features of the representative views for each model while SBR-2D–3D needs those features for all the sample views of each model. Thus, SBR-VC has better efficiency, especially when applied for large-scale retrieval applications.

Regarding the robustness to partiality, it is interesting to note that there is a slight improvement of the effectiveness measure when the partiality-based weights are used (see Tables 4, 5). As opposite to the SBR-2D–3D method, SBR-VC selects a more diverse set of representative views. Therefore, this set somehow helps to better discriminate difficult queries in comparison to the SBR-2D–3D method which is based on 2D views as well.

4.3 Partial shape retrieval using data-aware partitioning

We proposed a shape retrieval method for generic shapes based on the detection of interest points. The idea is to represent a shape using a global shape descriptor and a set of part descriptions. That is, given two 3D objects O and Q, we represent them as follows:

$$S_O = \{(s_O, P_O)|s_O \in R^n \text{ and } P_O = \{p_O^i\}_{i=1}^m, p_O^i \in R^n\},$$
$$S_Q = \{(s_Q, P_Q)|s_Q \in R^n \text{ and } P_Q = \{p_Q^i\}_{i=1}^k, p_Q^i \in R^n\},$$

where O has m partitions and Q has k partitions. The partitions are obtained by grouping Harris 3D keypoints [22] in the

Table 6 Performance measures of data-aware (without partiality)

Run	NN	FT	ST	MAP	MQR
$\mu = 0.8$	0.3431	0.2514	0.2100	0.2824	73.1792
$\mu = 0.9$	0.3457	0.2495	0.2088	0.2836	75.8807
$\mu = 1.0$	0.3406	0.2444	0.2053	0.2806	78.5606

Euclidean space using an adaptive clustering algorithm [23]. Subsequently, we compute the minimum enclosing sphere of the keypoints to define the partition. Finally, the mesh part contained in the sphere is considered a partition. The representations S_O and S_Q contain the description of the global shape (s_O and s_Q) and the descriptions of the partitions. In our experiments, we used the DESIRE descriptor proposed in [27] to describe the global shape and also the partitions.

To properly assess the similarity between two shapes, we need to define a distance $d(S_O, S_Q)$. This distance should measure the dissimilarity between two objects using their intermediate representations. We considered a linear combination between the global-to-global and the partition-based distance:

$$d(S_O, S_Q) = \mu \|s_O - s_Q\|_2 + (1 - \mu)d_{\text{part}}(P_O, P_Q), \quad (3)$$

where $0 \le \mu \le 1$ weights the contribution of the involved terms. At this point, we focus on the definition of an appropriate distance between two sets of partitions $d_{\text{part}}(P_O, P_Q)$. We proposed to formulate an objective matching function:

$$f(x) = \sum_{i,j} \|p_O^i - p_Q^j\|_2.x(i, j), \quad (4)$$

where $x(.)$ is a boolean indicator variable that indicates if p_O^i matches p_Q^j. The goal is to find the optimum x^* which minimizes $f(x)$. This can be formulated as an optimization problem to find the minimizer $x^* = \text{argmin}_x f(x)$, subject to $\sum_i x(i, j') = 1$ and $\sum_j x(i', j) = 1 \forall i, j$.

Finally, we define the distance function as

$$d_{\text{part}}(P_O, P_Q) = \frac{f(x^*)}{\min(|P_O|, |P_Q|)}. \quad (5)$$

where the normalization is to deal with partition sets of different lengths.

In this method, we test several values of μ to evaluate its impact on the final measured distance. For details about the configuration setup, please refer to [23]. Table 6 shows the results of data-aware method using the unweighted measures. Table 7 shows the results of data-aware method using the weighted measures.

The improvement in the use of part-based distance to a global approach is moderate in presence of partial data. The

Table 7 Performance measures of data-aware (with partiality)

Run	NN	FT	ST	MAP	MQR
$\mu = 0.8$	0.3364	0.2482	0.2081	0.2789	65.4082
$\mu = 0.9$	0.3387	0.2462	0.2068	0.2800	67.7836
$\mu = 1.0$	0.3336	0.2411	0.2033	0.2770	70.2277

Table 8 Performance measures of BoF (without partiality)

Run	NN	FT	ST	MAP	MQR
RSI (#$M = 300$)	0.0881	0.0727	0.0709	0.0914	159.6087
PSI (#$M = 100$)	0.0931	0.0809	0.0768	0.0968	153.1436
SC (#$M = 900$)	0.0872	0.0832	0.0775	0.0944	141.6518
FPFH (#$M = 300$)	0.1156	0.0788	0.0733	0.0965	155.6958

best MAP value is obtained for $\mu = 0.9$ which represents an improvement of 1.07 % with respect to the MAP obtained by only using a global descriptor ($\mu = 1.0$). Nevertheless, this finding was already discussed in [23] in the context of generic shape retrieval. In the case of partial data, this behavior seems to be accentuated by the difficulty of the queries. Obviously, it is more difficult to find representative partitions in partial data and match them to partitions in the target models. However, we believe that our method is able to detect representative partitions in objects with well-defined structure (see Sect. 5 for a detailed class-by-class evaluation).

With respect to the robustness to partiality, the results with the weighted performance measures present a slight drop. We believe that it was expected because our method strongly depends on global descriptors and partitions based on local features. Recall that partitions are determined by the distribution of keypoints on the mesh, so in presence of difficult queries, it is expected for these distributions to change considerably.

4.4 Partial shape retrieval using bag of features

The bag of features (BoF) approach resembles the organization of documents from a textual dictionary. In the multimedia literature, the idea is to find a feature dictionary. This dictionary is useful to find characteristic distributions of features which allow us to describe an object. In our framework, the BoF approach is applied in two steps. The first step consists of computing a dictionary of features using descriptors from the target set. Let D be the set of descriptors in $\mathbb{R}^n$ and k be the number of clusters we want to find. To find the dictionary, we use the k-means algorithm over the set of descriptors.

The set of centroids represents the dictionary (hereafter denoted as M). The second step of BoF approach consists of combining the set M and the local descriptors to compute a descriptor per shape. Let P be the set of local descriptors for one shape. We need to compute the feature distribution for each descriptor in P as $\theta(p_j) = (\theta_1(p_j), \ldots, \theta_k(p_j))^T$ where

$$\theta_i(p_j) = c(p_j)\exp\left(\frac{-\|p_j - m_i\|_2}{2\sigma^2}\right) \tag{6}$$

where $c(p_j)$ is a constant such that $\|\theta(p_j)\|_2 = 1$, m_i is the centroid of cluster C_i and σ is constant. Each bin in $\theta(p_j)$ represents the probability that descriptor p_j belongs to a cluster. Here we present the soft version of quantization as opposite to hard quantization where bins accounts for descriptors near to clusters. We use the soft version as it has shown to be effective in the shape retrieval domain [4].

The final descriptor for a shape represented by the set of descriptors P is computed as

$$f(P) = \sum_{p_j \in P} \theta(p_j) \tag{7}$$

and the matching between two objects can be performed using the L_1 distance between their quantized vectors. Nevertheless, in the context of whole-from-part retrieval, the Kullback–Leibler divergence has proven to be effective to compare quantized vectors in the BoF approach [17]. In all our experiments, we use the KL divergence as distance between a query and a target shape.

For our experiments, we compute the dictionary from descriptors of the target set. Subsequently, we use the dictionary to compute the descriptors for the target and the query set. In addition, in all our experiments, parameter σ was set to twice the median distance between centroids m_i.

To test the BoF approach, we use four different descriptors available in the Point Cloud Library [21]. To properly use the provided implementations, we transform the input mesh into a 3D point cloud. For this purpose, we sample 50,000 points on the surface using the sampling method proposed by Osada et al. [18]. Following we detail the configuration used for each descriptor:

- *Rectangular spin images (RSI)* [10] the image width was set to 8 and the radius search was set to $object\_diagonal \times 0.08$. Each spin image has a dimension of 153.
- *Polar spin images (PSI)* the configuration is the same as the rectangular spin images.
- *Shape contexts (SC)* [9] the maximal radius was set to $object\_diagonal \times 0.08$ and the minimal radius was set to 0.125 times the maximal radius. The dimension of the shape contexts is 1980.

Table 9 Performance measures of BoF (with partiality)

Run	NN	FT	ST	MAP	MQR
RSI (#$M = 300$)	0.0892	0.0734	0.0713	0.0917	156.3425
PSI (#$M = 100$)	0.0933	0.0812	0.0770	0.0972	149.7691
SC (#$M = 900$)	0.0861	0.0825	0.0771	0.0940	135.8128
FPFH (#$M = 300$)	0.1167	0.0799	0.0741	0.0971	153.2589

– *Fast point feature histogram (FPFH)* [19,20] the radius search was set to $object\_diagonal \times 0.08$. The dimension of the FPFH descriptor is 33.

We test several sizes for the vocabulary M. Nevertheless, due to the limited space, we only present the results with the best vocabulary sizes per descriptor type. Finally, for all experiments, we extract 300 Harris 3D keypoints [22] for each shape which are the input for the overall approach. Table 8 shows the results of BoF method using the unweighted measures. Table 9 shows the results of BoF method using the weighted measures.

In general in our evaluation, the BoF approach does not perform as well as previous approaches. In our opinion, quantization is not resilient to missing data, and therefore the distribution of local features in the queries is highly dissimilar to global shapes. However, it is worth noting that this method presents a moderate improvement with the evaluation of weighted measures (except for the shape context variation). In general, the use of local features is intended to provide robustness to missing data in some degree. Nevertheless, our dataset presents a difficult task even for approaches based on local features because the partial views do not exactly contain the same local geometry as the target models.

4.5 Partial shape retrieval with spin images and signature quadratic Form distance

This method involves the application of a flexible distance used to compare two shapes which are represented by feature sets. The signature quadratic form distance [1] is a context-free distance that has proven to be effective in the image retrieval domain. In addition, in this algorithm, we build a feature set composed of normalized local descriptors. The idea is to compute an intermediate representation for each shape using a set of local descriptors which are calculated around a set of representative surface points. This algorithm is a modified version of the method evaluated in [3].

First, we compute interest points using Harris 3D [22]. We select 2 % of the number of vertices of a shape (with the highest Harris response) as keypoints. In our experiments, in average the percentage ranges between 200 and 800 keypoints. These interest points are used as base points around

Table 10 Performance measures of SQFD (without partiality)

Run	NN	FT	ST	MAP	MQR
SQFD (SC)	0.2897	0.1842	0.1376	0.1712	97.3271
SQFD (SI)	0.3108	0.2043	0.1576	0.1978	84.5678

Table 11 Performance measures of SQFD (with partiality)

Run	NN	FT	ST	MAP	MQR
SQFD (SC)	0.3258	0.1925	0.1157	0.1848	67.6384
SQFD (SI)	0.3476	0.2086	0.1334	0.2034	61.4216

which the local descriptors will be computed. On the other hand, we use the complete set of vertices as accumulation points. If a shape has less than 50,000 vertices, our method samples points on the surface until reaching 50,000 points.

The set of local descriptors of a shape forms the feature space of that shape. Next, a local clustering algorithm [12] is applied to obtain a set of representative descriptors. In brief, the clustering uses two thresholds to define the inter-cluster and intra-cluster properties of the space, so it does not depend on the number of clusters. Hence, the clustering only depends on the distribution of the descriptors in the feature space. Given a partitioning after the clustering, the intermediate representation S^P of an object P is defined as a set of tuples as follows:

$$S^P = \{(c_i^P, w_i^P), i = 1, \ldots, n\} \tag{8}$$

where c_i^P is the average local descriptor in the i-th cluster and w_i^P is the fraction of elements belonging to the i-th cluster. It is worth noting that the representation of an object depends on the clustering and two objects do not necessarily have the same number of clusters.

For the experiments, we test two runs with different descriptors. The setup is described following:

– *Interest point detector* We use adaptive neighborhood around a vertex to compute the local support. Two percent of the number of vertex with the highest Harris response is selected as keypoints.
– *Spin images computation* Width of spin images $W = 25$, support angle $A_s = \pi$ and bin_size are set to the mesh resolution. These parameters allow us to compute spin images within a local support (a detailed description of these parameters can be found in [6]).
– *Shape contexts computation* We used the same configuration as presented in Sec. 4.4.
– *Clustering* We use 0.1 and 0.2 as intra-cluster and inter-cluster thresholds, respectively. The minimum number of elements per cluster was 10.
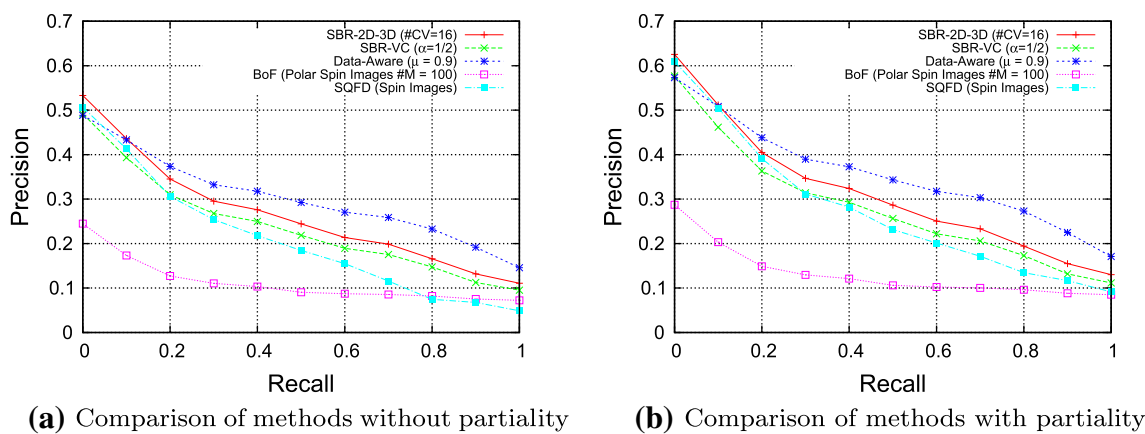
(a) Comparison of methods without partiality

(b) Comparison of methods with partiality

Fig. 7 Precision–Recall plots for the best configuration setup per method

– *SQFD* We use L_2 as ground distance and a Gaussian function with $\alpha = 0.9$ for the similarity function.

Tables 10 and 11 show the obtained results. From the results, we can note that spin images achieve a better performance than shape contexts. It is possible that the high dimensionality of shape contexts plays an important role in the local clustering for computing the signatures. The higher the dimensionality, the more difficult is to find the well-defined clusters in the distribution of local features in a shape. Therefore, this causes signatures to be flat in general and hence they are not so representative.

In contrast to the BoF approach, the SQFD method only depends on the local clustering of local features in a shape. We believe that the improvement achieved by SQFD over BoF reveals that the visual dictionary (which is based in the target models) do not represent the information of partial queries. In contrast, SQFD exploits the local distribution of features and therefore the signatures are more representative. As the final signature is found by averaging the local distributions, it is possible to obtain a more stable representation for both target models and partial queries (somehow dealing with outliers). It is also interesting to note that the use of local features allows to obtain an improvement when the weighted measures are used.

5 Evaluation and discussion

In this section, we make a comparison of the best runs of each presented method in Sect. 4. We chose the runs with the best MAP and compare them using the standard measures and the measures with partiality weights. Figure 7; Tables 12 and 13 show the comparison of the best runs.

There are two aspects to remark from these results. First, SBR-2D–3D approach obtains the best result to retrieve the most similar shape to a partial query. Note how the SBR-

Table 12 Performance measures of the best configurations (without partiality)

| Method	NN	FT	ST	MAP	MQR
SBR-2D–3D (#CV=16)	**0.3535**	0.2290	0.1808	0.2455	**62.7326**
SBR-VC ($\alpha = 1/2$)	0.3218	0.2065	0.1638	0.2199	75.3940
Data-aware ($\mu = 0.9$)	0.3457	**0.2495**	**0.2088**	**0.2836**	75.8807
Polar spin images (#$M = 100$)	0.0931	0.0809	0.0768	0.0968	153.1436
SQFD (spin images)	0.3108	0.2043	0.1576	0.1978	84.5678

The higher performances for each evaluation measure are in bold

Table 13 Performance measures of the best configurations (with partiality)

Run	NN	FT	ST	MAP	MQR
SBR-2D–3D (#CV=16)	**0.3504**	0.2279	0.1803	0.2447	**56.7151**
SBR-VC ($\alpha = 1/2$)	0.3209	0.2066	0.1639	0.2202	70.9168
Data-aware ($\mu = 0.9$)	0.3387	**0.2462**	**0.2068**	**0.2800**	67.7836
Polar spin images (#$M = 100$)	0.0933	0.0812	0.0770	0.0972	149.7691
SQFD (spin images)	0.3476	0.2086	0.1334	0.2034	61.4216

The higher performances for each evaluation measure are in bold

2D–3D method obtains the best NN and the best MQR consistently in both kinds of evaluation (unweighted and weighted). More specifically, regarding the unweighted measures (Table 12) SBR-2D–3D obtained an improvement of 2.25% and 17% with respect to NN and MQR against the second best method in each measure, respectively. This means that SBR-2D–3D has the ability of retrieving a relevant object or the exact object for a partial query with a good chance. We believe that this behavior is due to the exploration of a dense set of views to perform the matching (81 in our experiments). It is likely that among the dense set of views, there is one which is very similar to the view obtained with the projection method from the icosahedron faces. This result is also

(a) Class vs. MAP

(b) Class vs. MAP
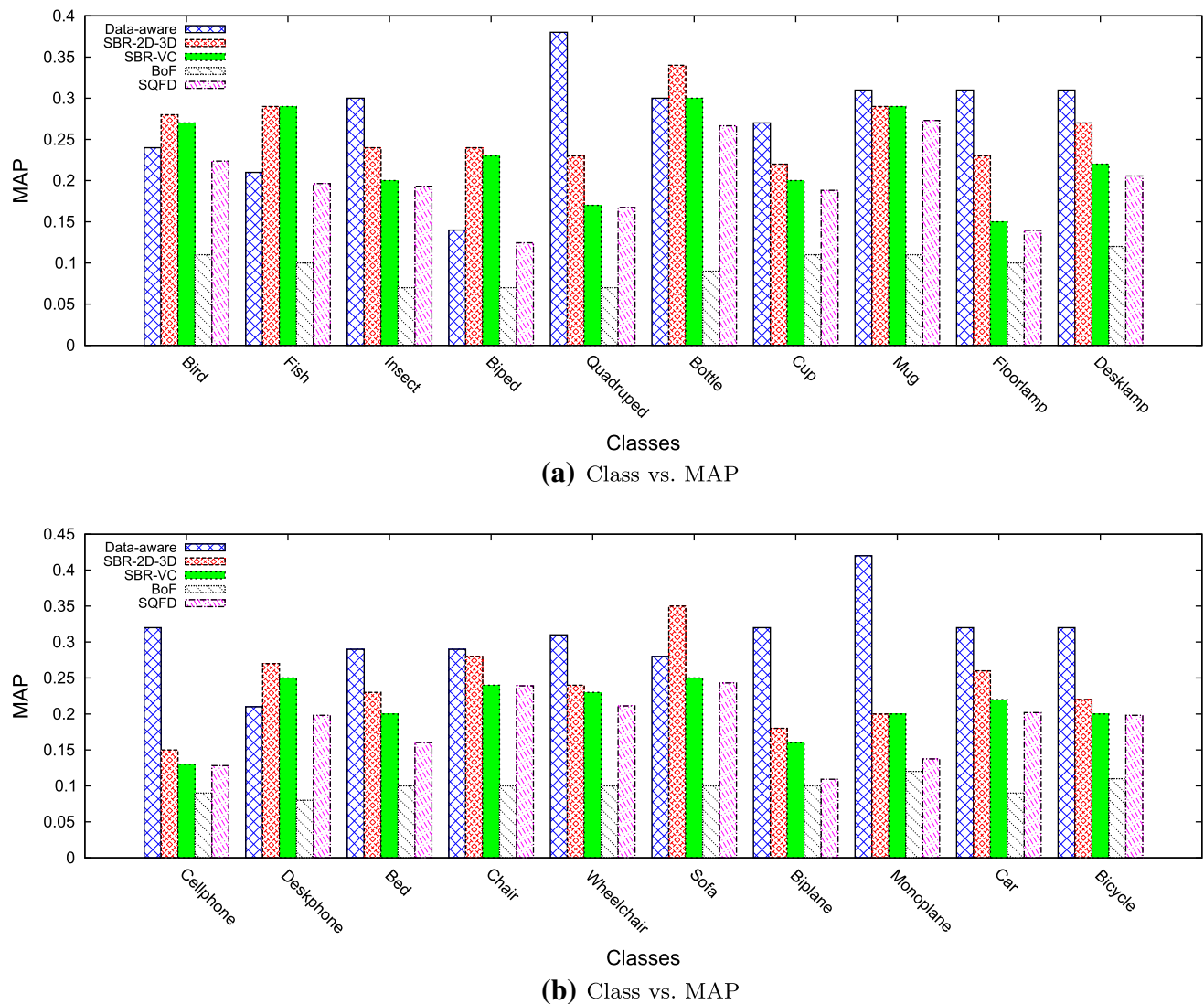
Fig. 8 Mean average precision per class for the evaluated methods

observed in the precision–recall plots in Fig. 7 where SBR-2D–3D obtains the highest values of precision for small values of recall. Note that small values of recall are very related to the nearest neighbor measure. Furthermore, the analysis is similar if we consider the weighted measures (Table 12): SBR-2D–3D obtained an improvement of 3.45 and 8 % in NN and MQR, respectively. The drop in the MQR improvement when considering the unweighted and weighted measures can be attributed to the same analysis provided in Sect. 4.1: SBR-2D–3D fails to retrieve similar models when the query is challenging.

Second, the data-aware method obtains the best results when the evaluation is done over the complete list of retrieved objects. That is, this method is able to measure similarity between a partial view and target objects in the same class. This can be shown in the best FT, ST and MAP obtained (from Fig. 7, note also how the precision–recall curve for

this method outperforms the others for recall values >0.2). In addition, regarding the unweighted measures, data-aware obtained an improvement of 8.95, 15.45 and 15.51 % with respect to FT, ST and MAP against the second best method, respectively. A reason for these results is that the partitioning scheme is consistent in queries and target shapes in the same class, and therefore is a good representation to preserve the intra-class similarity. We also believe that the application of global descriptors is particularly useful in this context because there are views which are very similar to the target models when both are normalized in pose (the DESIRE method includes a pose normalization step prior to the description). In our opinion this is not a generalized rule, but when it occurs our method exploits the global similarity in conjunction with the partition-based matching to enhance the structure of the models. Again, the analysis is similar when the weighted measures are used. Data-aware

obtained an improvement of 8.02, 14.69 and 14.42 % with respect to FT, ST and MAP against the second best method, respectively. It is also worth noting that there is a drop in the improvements of the measures when we use the weighted measures. This fact is in accordance with the analysis provided in Sect. 4.3 in that data-aware also fails to retrieve relevant models when the query is challenging. Nevertheless, it is also important to remark that the drop in improvement is slight.

We also performed a experiment to evaluate the behavior of each method with respect to each class. Note that all the measures we use so far are averaged over the complete set of queries. For this reasons, it is possible that those results are hiding some important information to have a clue about how each method works. Figure 8 shows the MAP averaged by class for each evaluated method. An important clue that can be extracted from this result is that view-based approaches are complementary to the data-aware method. There is a notable improvement of SBR-2D–3D against data-aware in classes such as Fish, Bottle and Sofa. Interestingly, objects from these classes shares a characteristic: they contain large smooth surfaces. Therefore, it is possible that partial queries coming from these classes contain a large portion of the real object. Apparently, this fact is being exploited by the view-based approaches, more specifically SBR-2D–3D.

Following the same logic, we identified the classes in which data-aware outperforms the other methods. A notable improvement can be observed in classes such as Insect, Quadruped, Floorlamp, Cellphone, Biplane and Monoplane. Interestingly, object from these classes also share a characteristic: they are composed by well-defined parts and small variation structure within the class. The small variation is being exploited by the global descriptor while the partitioning takes advantage of the structure.

As a result, we believe that the two representations (view-based and partition-based) convey complementary information, and therefore we can exploit the best of both worlds to address the problem of partial shape retrieval. In this direction, we explored the impact of combining these two approaches in a naive way to further improve the results obtained so far independently for each method.

We performed an experiment which considers to combine the two best methods by means of a simple combination rule of the provided distance matrices. For the results presented in this section, we took the distance matrices of SBR-2D–3D and data-aware to produce new distance matrices associated to a combination weight ω. Let $distSBR$ be the distance matrix of the best run of SBR-2D–3D and $distAware$ be the distance matrix of the best run of data-aware, we combine the distance matrices using $combinedDistance = \omega \cdot distAware + (1-\omega) \cdot distSBR$. Then, the new combined distances were evaluated as usual.

Table 14 Performance measures for combination SBR-2D–3D + data-aware (without partiality)

Weight	NN	FT	ST	MAP	MQR
$\omega = 0.2$	**0.4272**	0.2930	0.2296	0.3222	**40.4988**
$\omega = 0.3$	0.4217	0.2975	0.2366	**0.3316**	41.4051
$\omega = 0.4$	0.4094	**0.2978**	**0.2403**	0.3308	45.0901

The higher performances for each evaluation measure are in bold

Table 15 Performance measures for combination SBR-2D–3D + data-aware (with partiality)

Weight	NN	FT	ST	MAP	MQR
$\omega = 0.2$	**0.4232**	0.2914	0.2286	0.3205	**33.1614**
$\omega = 0.3$	0.4170	0.2957	0.2357	**0.3295**	34.1445
$\omega = 0.4$	0.4045	**0.2958**	**0.2391**	0.3284	37.9183

The higher performances for each evaluation measure are in bold

Tables 14 and 15 show the results of combinations. It is worth noting the improvement of all measures (without and with partiality weights) compared to isolated methods. Here, the best MAP achieved is 0.3316, which represents a notable increment (17 %) with respect to the best MAP obtained (0.2836) in the comparison of the previous section. Also, it is important to note the influence of both methods in the combination. For example, when the combination contains high contribution of SBR-2D–3D ($\omega = 0.2$), we obtain the best NN and MQR. This is consistent with our observation that SBR-2D–3D is suitable to obtain the best target shape for a partial query. On the other hand, the best retrieval results occur with more balanced contributions of both methods ($\omega = 0.3$ and $\omega = 0.4$).

These results show the ability of each method to perform similarity search with partial 3D shapes. In addition, it is worth noting that different approaches contribute in different ways to the retrieval task. Therefore, it seems that the approaches compute complementary representations, which are able to get different aspects of partial queries. This is the reason why the combination of the two best methods performs better than the isolated methods.

Timing

An important aspect to evaluate is the efficiency of methods for partial shape retrieval. Table 16 shows the average time for each algorithm to perform a similarity search given a partial view as query. The platform used in SBR-2D–3D and SBR-VC was composed of a DELL Precision T7500 machine with an Intel Xeon CPU X5675 @3.70 GHz 3.06 GHz (2 processors, 12 cores), 20GB memory and Windows 7 64-bit OS. On the other hand, Data-Aware, BoF and SQFD were evaluated on a Intel Core i7-3537U processor @2.00GhZ

Table 16 Query times for the evaluated methods

Method	Setup	Query times (sec)
SBR-2 D-3D	#CV = 16	1.9987
	#CV = 8	1.1119
	#CV = 4	0.4932
SBR-VC	$\alpha = 1/2$	1.6893
	$\alpha = 1/3$	1.4949
Data-aware	any setup	3.06
BoF	RSI (#M = 300)	0.3008
	PSI (#M = 100)	0.2145
	SC (#M = 900)	0.5412
	FPFH (#M = 300)	0.2412
SQFD	SC	1.8842
	SI	1.5687

(4 cores) with 8GB and Linux OS. Platforms are different because the evaluation was done separately by two different teams as part of a SHREC contest [24]. Still, we believe that the presented results are useful to compare the evaluated methods.

As we can see, the query time of the view-based methods depends on the number of views used for assessing the similarity. For example, reducing the number of views from 16 to 4 in SBR-2D–3D reduces the query time in almost a quarter. We can see a similar behavior in SBR-VC, but the improvement is less notorious, probably because the query time is dominated by the view clustering. It is important to recall that more views provided the best effectiveness results, therefore there is a trade-off between effectiveness and efficiency depending on the number of views used in the distance measure.

On the other hand, data-aware does not depend on the parameter for the combination of global and part description. This is because the query time is completely dominated by the optimization for the matching (computation of keypoints and partition took in average 0.4 s). Again, note that data-aware was the most effective method, so its application depends on either effectiveness which is most important to the expenses of computational time.

Interestingly, the BoF approach obtained the lower query times. Basically, these times include point sampling, description, quantization and query time. Nevertheless, it is worth noting that the BoF approach requires a off-line step to compute the vocabulary. In our experiments, this step took 5–10 h depending on the feature dimension and the number of cluster for the dictionary. Once we have computed the vocabulary, the query process is very fast.

Finally, SQFD presents a intermediate query time between Data-Aware and BoF. Most of the time in SQFD method is devoted to the computation of the SQFD distance which is expensive (computation of keypoints, description and local clustering took in average 0.21 s).

6 Conclusions and future work

In this paper, we evaluated five methods using a large-scale dataset with simulated partial views. The dataset is composed of a set of partial views generated based on a target set of shapes. To the best of our knowledge, this is the first attempt to evaluate partial shape retrieval algorithms in a large-scale scenario. In addition, we introduced a novel-weighted performance measure which involves the complexity and difficulty of the queries.

Our results show that the dataset was very challenging. Firstly, although the combination of methods showed to improve the results compared to the evaluation of the independent methods, the results still remain moderate. This is an indication that the problem is far from being solved. Moreover, in our opinion, the dataset represents a scenario for real-world applications because it was built by simulating the real scanning process. Therefore, it is important to realize this in to find out the real capabilities of existing algorithms. Secondly, the combination of approaches seems to be the direction to find new solutions to the whole-from-part retrieval problem. However, we believe that more sophisticated combinations of complementary descriptions need to be evaluated. We plan to go in this direction in the feature, trying to obtain complementary representations for shapes and partial queries. Thirdly, efficiency and robustness issues do matter. Obviously, for large-scale retrieval tasks, it is necessary to have fast algorithms which are able to deal with imperfections of meshes obtained from real devices. As a consequence, we identify robust partial shape retrieval algorithm scalable to large datasets as a promising future research direction. We identify additional interesting future work for the generation of even more realistic retrieval benchmarks. In particular, one may wish to control the level of resolution of the acquisition process, or introduce various kinds of data noises. In particular, varying lighting conditions and reflectance properties that influence the precision degrees of 3D acquisition, could be considered.
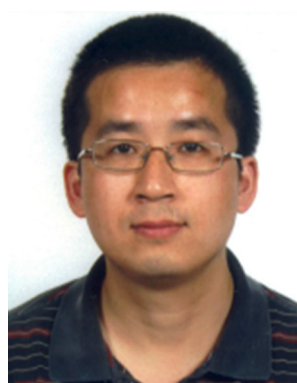
References

1. Beecks, C., Uysal, M.S., Seidl, T.: Signature quadratic form distances for content-based similarity. In: Proceedings of the 17th ACM International Conference on Multimedia, MM'09, pp. 697–700. ACM (2009)

2. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. **24**(4), 509–522 (2002)

3. Biasotti, S., Bai, X., Bustos, B., Cerri, A., Giorgi, D., Li, L., Mortara, M., Sipiran, I., Zhang, S., Spagnuolo, M.: SHREC'12 Track: Stability on Abstract Shapes. In: Proc. Eurographics Workshop on 3D Object, pp. 101–107. Eurographics Association (2012)

4. Bronstein, A., Bronstein, M., Guibas, L., Ovsjanikov, M.: Shape Google: geometric words and expressions for invariant shape retrieval. ACM Trans. Comput. Graph. 30(1), 1–20 (2011)

5. Browatzki, B., Fischer, J., Graf, B., Bulthoff, H., Wallraven, C.: Going into depth: evaluating 2D and 3D cues for object classification on a new, large-scale object dataset. IEEE ICCV Workshops **2011**, 1189–1195 (2011)

6. Bustos, B., Sipiran, I.: 3D shape matching for retrieval and recognition. In: Pears, N., Liu, Y., Bunting, P. (eds.) 3D Imaging, Analysis and Applications, pp. 265–308. Springer, London (2012)

7. Dutagaci, H., Godil, A., Axenopoulos, A., Daras, P., Furuya, T., Ohbuchi, R.: SHREC'09 Track: Querying with Partial Models. In: Proc. Eurographics Workshop on 3D Object Retrieval, pp. 69–76. Eurographics Association (2009)

8. Dutagaci, H., Godil, A., Cheung, C.P., Furuya, T., Hillenbrand, U., Ohbuchi, R.: SHREC'10 Track: Range Scan Retrieval. In: Proc. Eurographics Workshop on 3D Object Retrieval, pp. 109–115. Eurographics Association (2010)

9. Frome, A., Huber, D., Kolluri, R., Bulow, T., Malik, J.: Recognizing objects in range data using regional point descriptors. In: Proceedings of the European Conference on Computer Vision (ECCV) (2004)

10. Johnson, A.: Spin-images: a representation for 3-D surface matching. Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh (1997)

11. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view RGB-D object dataset. In: ICRA, pp. 1817–1824. IEEE (2011)

12. Leow, W.K., Li, R.: The analysis and applications of adaptive-binning color histograms. Comput. Vis. Image Underst. **94**, 67–91 (2004)

13. Li, B., Johan, H.: View context: a 3D model feature for retrieval. In: Boll, S. et al. (eds.): MMM 2010, LNCS, pp. 185–195. Springer, Heidelberg (2010) (5916)

14. Li, B., Johan, H.: Sketch-based 3D model retrieval by incorporating 2D–3D alignment. Multimed. Tools Appl. **65**(3), 363–385 (2013)

15. Li, B., Lu, Y., Johan, H.: Sketch-based 3D model retrieval by viewpoint entropy-based adaptive view clustering. In: Proc. Eurographics Workshop on 3D Object Retrieval, pp. 49–56. Eurographics Association (2013)

16. Li, B., Schreck, T., Godil, A., Alexa, M., Boubekeur, T., Bustos, B., Chen, J., Eitz, M., Furuya, T., Hildebrand, K., Huang, S., Johan, H., Kuijper, A., Ohbuchi, R., Richter, R., Saavedra, J.M., Scherer, M., Yanagimachi, T., Yoon, G.J., Yoon, S.M.: SHREC'12 track: sketch-based 3D shape retrieval. In: Proc. Eurographics Workshop on 3D Object Retrieval, pp. 109–118. Eurographics Association (2012)

17. Liu, Y., Zha, H., Qin, H.: Shape topics: a compact representation and new algorithms for 3d partial shape retrieval. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2006)

18. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Shape distributions. ACM Trans. Gr. **21**(4), 807–832 (2002)

19. Rusu, R., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3D registration. In: Robotics and Automation, 2009. ICRA '09. IEEE International Conference, pp. 3212–3217 (2009)

20. Rusu, R., Holzbach, A., Blodow, N., Beetz, M.: Fast geometric point labeling using conditional random fields. In: Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference, pp. 7–12 (2009)

21. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: International Conference on Robotics and Automation. Shanghai (2011)

22. Sipiran, I., Bustos, B.: Harris 3D: a robust extension of the Harris operator for interest point detection on 3D meshes. Vis. Comput. **27**(11), 963–976 (2011)

23. Sipiran, I., Bustos, B., Schreck, T.: Data-aware 3D partitioning for generic shape retrieval. Comput. Gr. **37**(5), 460–472 (2013)

24. Sipiran, I., Meruane, R., Bustos, B., Schreck, T., Li, B., Lu, Y., Johan, H.: SHREC'13 Track: Large-Scale Partial Shape Retrieval Using Simulated Range Images. In: Proc. Eurographics Workshop on 3D Object Retrieval, pp. 81–88. Eurographics Association (2013)

25. Takahashi, S., Fujishiro, I., Takeshima, Y., Nishita, T.: A feature-driven approach to locating optimal viewpoints for volume visualization. In: IEEE Visualization, p. 63. IEEE Computer Society (2005)

26. Veltkamp, R., Ter Haar, F.: SHREC 2007 3D retrieval contest. Technical report. Department of Information and Computing Science (2007)

27. Vranic, D.V.: DESIRE: a composite 3D-shape descriptor. In: Proceedings of the IEEE International Conference on Multimedia and Expo (2005)
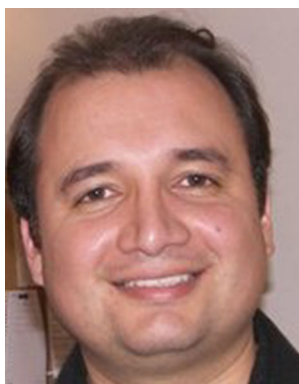
Ivan Sipiran is a Post-doc researcher in the Department of Computer and Information Sciences at the University of Konstanz. Currently, he is a researcher associated to the EC FP7 STREP Project PRESIOUS. He got a Ph.D. in Computer Science at the University of Chile in 2014. His research interests include 3D object retrieval, geometry processing and computer vision.
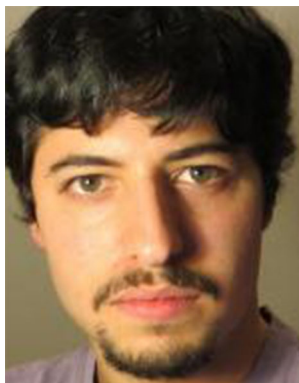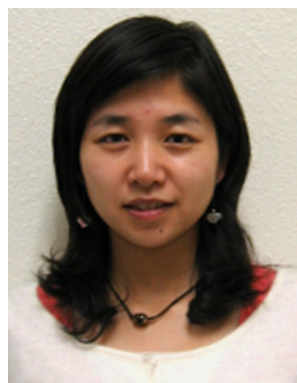
Rafael Meruane is a master student at the Department of Computer Science, University of Chile. His research interests include computer vision and 3D object retrieval.

Benjamin Bustos is an Associate Professor at the Department of Computer Science, University of Chile. He is head of the PRISMA Research Group. He leads research projects in the domain of content-based multimedia information retrieval. His research interests include similarity search, 3D object retrieval, multimedia mining, metric/nonmetric indexing, and pattern recognition. Benjamin Bustos obtained a doctoral degree in natural sciences from the University of Konstanz, Germany, in 2006.

Tobias Schreck is an Assistant Professor for Visual Analytics with the Department of Computer and Information Science at the University of Konstanz, Germany. His research interests include visual search and analysis of time-oriented, high-dimensional, and 3D object data, with applications in data analysis and multimedia retrieval. Tobias Schreck received a PhD in Computer Science from the University of Konstanz in 2006. Tobias Schreck has served as a workshop co-chair for the EG 3D Object Retrieval workshop series, and regularly assumes committee roles for the IEEE VIS conference series. Tobias Schreck at present is a Principal Investigator in the EU Research Projects PRESIOUS and CONSENSUS as well as a research project on Visual Analytics sponsored by the Assistant Professor Program of the Baden-Württemberg Foundation.

Bo Li is a Postdoctoral Research Associate in Texas State University. He received his PhD and MS degrees in Computer Science from Nanyang Technological University (Singapore) in 2012, and Xi'an Jiaotong University (China) in 2005, respectively. Previously, he was a Guest Researcher in National Institute of Standards and Technology. His research interests include 3D model retrieval, shape matching, and 3D modeling.

Yijuan Lu is an Assistant Professor at Computer Science Department, Texas State University. Her research interests include Multimedia, Computer Vision, and Machine Learning. She was the Best Paper Award recipient of ICME 2013 and ICIMCS 2012. Her research has been supported by NSF, DoD, Army Research, TxDOT, and Texas State.

Henry Johan is a Senior Research Fellow in Fraunhofer IDM@NTU (Singapore). Previously, he was a Post-Doctoral Fellow in the Department of Complexity Science and Engineering at the University of Tokyo (Japan). Then, he joined the School of Computer Engineering at Nanyang Technological University (Singapore) as an Assistant Professor. His research interests in computer graphics include rendering, animation, and shape retrieval. He received his BS, MS, and PhD degrees in Computer Science from the University of Tokyo in 1999, 2001, and 2004, respectively.