Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

ShapeBench: a new approach to benchmarking local 3D shape descriptors

Bart Iver van Blokland[1,]

ARTICLE INFO

Article history:
Received July 22, 2024

Keywords: 3D Local Shape Descriptors, ShapeBench, Benchmark

ABSTRACT

The ShapeBench evaluation methodology is proposed as an extension to the popular Area Under Precision-Recall Curve (PRC/AUC) for measuring the matching performance of local 3D shape descriptors. It is observed that the PRC inadequately accounts for other similar surfaces in the same or different objects when determining whether a candidate match is a true positive. The novel Descriptor Distance Index (DDI) metric is introduced to address this limitation. In contrast to previous evaluation methodologies, which identify entire objects in a given scene, the DDI metric measures descriptor performance by analysing point-to-point distances. The ShapeBench methodology is also more scalable than previous approaches, by using procedural generation. The benchmark is used to evaluate both old and new descriptors. The results produced by the implementation of the benchmark are fully replicable, and are made publicly available.

1. Motivation

The ability to compare the similarity of 3D surfaces is crucial in a number of applications, such as 3D registration [1], bin picking [2], Simultaneous Localisation and Mapping (SLAM) [3] and 3D object retrieval [4]. A wide variety of methods have been proposed, both in the form of traditional algorithms [5, 6] and, more recently, learned features [7, 8]. Evaluating the performance of 3D surface matching methods provides understanding of their strengths and weaknesses, and is thus crucial for determining their practical applicability.

This paper focuses on improving the Precision Recall Curve (PRC)—the most popular methodology for evaluating local 3D shape descriptor methods—along with its associated Area Under Curve (AUC_{pr}) metric [9, 10, 11]. However, its application domain extends to any surface point matching algorithm. The AUC_{pr} metric measures the extent to which a particular method can correctly identify models in a set of scenes. The set of models M contains known objects. Subsets of models are placed in different arrangements, and exposed to various adverse conditions, to construct the set of scenes S. A detailed description of how the PRC and AUC_{pr} are calculated is given in Section 2.1.

We observe several issues with the PRC methodology, and how it is used to evaluate local descriptor methods in previous work. Most pertinently, the PRC methodology assumes that each surface point in a scene has at most one matching surface point on exactly one specific model. This does not adequately account for the possibility of multiple matches to exist, which can be caused by the presence of self-similarity within a model, or different models containing partially similar geometry. The methodology can therefore count true positives as false positives. An example of an object commonly used in previous evaluations exhibiting self-similarity is shown in Figure 1. Computing a ground truth of all surface matches is computationally intractable.

The ShapeBench methodology is proposed to address this, which—in contrast to previous work—measures the matching performance of a local 3D shape descriptor by comparing distances between individual descriptor pairs. The descriptors are computed for corresponding points on a model, and a modified version of the same model that constitutes the scene. Using this approach avoids the need to compute all ground truth matches,

e-mail: bart.van.blokland@ntnu.no (Bart Iver van Blokland)

Fig. 1: All surfaces that are similar to the indicated point in the *Armadillo* model from the Stanford 3D scanning repository according to the RICI descriptor. A darker red colour indicates a better match.

as the same point on two variants of the same surface is always a known match. The novel Descriptor Distance Index (DDI) metric is also proposed as a means to contextualise the computed distances between descriptor pairs, by quantifying the degree to which one descriptor can distinguish its counterpart from noise. The DDI is intended to be used in conjunction with the PRC, and can assist in explaining observed performance by the PRC, and visualise how performance degrades when a method misidentifies a nearest neighbour point.

Another motivation for comparing a model against a modified counterpart is that it improves the scalability of the evaluation methodology. Previous work has commonly relied on datasets of captured 3D data. The results presented in this article show that the quantity and variety of objects in these datasets is likely insufficient. Table 1 contains an overview over various recently proposed methods, along with all datasets used to evaluate them. These datasets contain at most 8 different models, and several also share the same model set.

The issue of quantity can in theory be rectified by using larger, more varied, datasets. However, datasets consisting of real world 3D scans scale poorly. Each scene must be constructed, captured, and stored separately. Multiple datasets available today containing only single objects require more than a terabyte to store, such as the ABC and Objaverse datasets listed in Table 1. An associated set of scenes would require a multiple of that. The proposed ShapeBench methodology therefore constructs scenes in a procedural and replicable manner, only requiring a set of models as input. Scenes are generated using a sequence of one or more *filters*, each simulating real world adverse conditions.

The use of artificial data has the added benefit that the effect of different adverse matching conditions can be studied in isolation. Real data often inherently contains combinations of these. One downside of artificial data is that multiple effects that naturally occur in real scans must now be approximated or simulated through one or more filters instead.

2. Related work

An overview is provided over different metrics and evaluation methodologies that have been used in previous work, with a special focus on the PRC methodology. A brief description is also given of the methods that were used to test the proposed ShapeBench benchmark in Section 5.

2.1. The PRC methodology

An overview over the procedure for computing the PRC and the associated AUC metric is given here. Because implementation details of the PRC vary, the version described by Guo et al. [5] is used as a reference.

For computing the PRC, a set of points $P_S \subseteq S$ is randomly sampled from the surface of the scene S, which may be done using a keypoint detector. Using known ground truth transformations, another set $P_M = \{T(q) : q \in P_S$ is constructed of model surface points that correspond to those in P_S, where $T(q)$ is the ground truth transformation that transforms the point q into the coordinate space of the model it belongs to.

After computing a feature vector for each point in P_S and P_M using the method being tested, the closest two points in feature space p_{m1} and p_{m2} are found in P_M for each point p_s in P_S. Using these, the nearest neighbour distance ratio, σ, is defined in Equation 1, where $f(p)$ denotes a feature vector for a given point p, and $d(f_1, f_2)$ a function computing the distance between two feature vectors.

$$\sigma = \frac{d(f(p_s), f(p_{m1}))}{d(f(p_s), f(p_{m2}))} \tag{1}$$

If the value of σ is below a threshold τ, the point pair p_s and p_{m1} is considered a match. For the point to be counted as a true match, two conditions must also be satisfied. Condition 1 requires that both points correspond to the same object, and condition 2 that the Euclidean distance between $T(p_s)$ and p_{m1} is less than half of the support radius. The support radius of a local shape descriptor is a parameter that determines the size of the support volume, usually a sphere or cylinder. All surfaces within this volume are represented by the descriptor. If either of these conditions is not satisfied, the pair is instead considered a false positive.

The PRC is computed by first computing the values of σ, and the two criteria, for each corresponding point pair in P_S and P_M. Varying the value of τ between 0 and 1, and computing the Precision and Recall for each point p_s, yields the PRC curve. Precision and Recall are defined in equations 2 and 3, respectively. The area below this curve constitutes the derived Area Under Curve (AUC_{pr}) metric.

$$Precision = \frac{|\text{true matches}|}{|\text{true matches}| + |\text{false positives}|} \tag{2}$$

$$Recall = \frac{|\text{true matches}|}{|\text{corresponding ground truth points}|} \tag{3}$$

The distance threshold τ was initially proposed by Lowe [26] for determining whether a database of keypoints contained a good match for a given query. Distinctive descriptors tend to only have a single good nearest neighbour, causing the σ ratio to be low. The threshold was later adopted for evaluating 3D descriptors [27, 28].

Dataset Information				Used in Evaluation					
Dataset	**Model Set**	**Models**	**Scenes**	[10]	[12]	[9]	[13]	[14]	[11]
Bologna 3D Retrieval (B3R) [15]	Stanford	6	18	Yes	-	Yes	Yes	Yes	Yes
Random Views [15]	Stanford	6	36	-	-	Yes	-	-	-
Bologna Dataset 1&2 - Stanford [16]	Stanford	6	45	-	Yes	Yes	-	-	-
UWA 3D Modelling	UWA	4	75	Yes	-	-	*Yes*[1]	Yes	-
UWA Object Retrieval [17, 18]	UWA	5	50	Yes	Yes	-	Yes	Yes	Yes
Bologna Dataset 3 - SpaceTime Stereo [16]	Kinect (+ clutter)	8	15	-	-	Yes	-	-	Yes
Bologna Dataset 5 - Kinect [19]	Kinect (+ clutter)	6	16	Yes	-	*Yes*[1]	-	-	-
Bologna Object Recognition	Kinect (+ clutter)	6	17	-	Yes	-	-	-	-
Bologna Mesh Registration	Kinect	6	95	Yes	-	-	Yes	-	-
Queens LiDAR [20]	Queens	5	63	-	Yes	-	-	-	-
7-scenes [21]	7-scenes	7	n/a	-	-	Yes	-	-	-
DTU [22]	DTU	45	3,204	-	-	-	-	-	-
ShapeNetCore [23]	ShapeNetCore	51,300	n/a	-	-	-	-	-	-
ABC [24]	ABC	1,000,000	n/a	-	-	-	-	-	-
Objaverse [25]	Objaverse	798,759	n/a	-	-	-	-	-	-

Table 1: An overview over datasets used for the evaluations in a number of recent papers, as well as some examples of larger datasets. Datasets that were *not* used in a particular evaluation are marked with a hyphen (-) for visual clarity. All datasets with equivalent *model set* names use the same (sub)set of models.

2.2. Evaluation methodologies

While only the PRC methodology has been discussed in detail, it is not the only one which has been used for evaluating descriptors to date. We therefore highlight some other notable metrics here.

A classic metric is the Receiver Operating Characteristics (ROC), developed during the second world war to evaluate the performance of radar operators. This metric plots the true positive rate against the false positive rate, where the true positive rate is equivalent to the recall metric in Equation 3. While not a common occurrence, the metric has seen use in the form of a confusion matrix [29, 30, 17]. The area under the ROC curve can be used as an aggregate metric for the overall performance of a tested method [31], in a similar fashion to the PRC..

Another metric that has been used is the Cumulative Match Characteristic (CMC) [7, 32], which uses a fixed number of query descriptors and their corresponding lists of nearest neighbours in feature space to compute the fraction where the ground truth nearest neighbour is in the top n nearest neighbours. The fraction is subsequently plotted for varying values of n. Van Blokland et al. use a variation of this metric [33, 34], computing the CMC solely for $n = 0$, and plotting the variation of its value across a number of scenes.

Whereas the PRC approach uses the area underneath the precision-recall curve to compute an overall performance metric, Buch et al. instead used the maximum F1 score [35], defined as the maximum harmonic mean across all the computed precision-recall values.

2.3. Local 3D Shape Descriptors

3D descriptors are commonly classified into global and local descriptors. Global descriptors aim to represent an entire model in a single descriptor. This has a clear space advantage over local descriptors, which use many descriptors to represent smaller portions of an object. However, local descriptors tend to be less sensitive to challenging matching conditions such as occlusion [36]. They also rely on an object being segmented from the environment. Examples include the SSCD [37] and PANORAMA [38] descriptors. Local descriptors are often combined with keypoint detectors to first locate distinct points of interest in a scene in order to reduce the volume of descriptors that need to be computed and compared.

One of the earlier local 3D shape descriptors is the Spin Image [39], proposed by Johnson and Hebert. The descriptor is a histogram that computes the distribution of points in the cylindrical coordinate space described by a given keypoint and its associated normal vector. Tombari et al. proposed the Unique Shape Context (USC) [16], which uses a spherical support volume subdivided into partitions along the azimuth, elevation, and radial directions. A histogram is subsequently computed over the surface points in the support volume that fall into each bin, scaled by the local density of each point. The method is an extension to the 3D Shape Context [40], and addresses its primary limitation by using a local reference frame to orient the support volume of the descriptor in a repeatable manner.

The Signature of Histograms of OrienTations (SHOT) [19] proposed by Salti et al. uses the same local reference frame and spatial subdivision of its support volume as the USC descriptor. In contrast to USC, SHOT accumulates histograms of cosines for each spatial bin. These cosines are computed between the normal vectors of surface points and the orientation of the descriptor.

The Rotational Projection Statistics (RoPS) descriptor proposed by Guo et al. [41] also uses a local reference frame to orient the points present in the support volume. These points are subsequently rotated in several increments along each major axis. For each rotated point cloud, all points are projected on the xy, yz, and xz planes, and a histogram is computed over their

[1]The information provided in the paper was insufficient to accurately deduce which exact dataset was used. A best guess has been used instead.

distribution. Various statistics are computed and concatenated to form the RoPS descriptor.

The Radial Intersection Count Image (RICI) descriptor proposed by van Blokland and Theoharis [33] is a histogram utilising variations in the number of intersections between circles and the object surface. The same authors noticed that intersection counts do not vary between most adjacent circles, and therefore proposed a more compact binary version of the descriptor, called the Quick Intersection Count Change Image (QUICCI) [34].

Other relevant examples of local 3D shape descriptors include the Fast Point Feature Histogram (FPFH) [42], and CoSPAIR [43].

3. The ShapeBench benchmark

The proposed ShapeBench evaluation methodology is now presented. Its objective is to evaluate whether a descriptor is capable of correctly determining similarity in surface point pairs, and to what extent this capability is maintained when presented with various adverse conditions that are common in practical applications. In order to achieve this, it is necessary to establish ground truth matches between surface points and their surrounding surface patches. While metrics exist for determining the similarity of such patches, exhaustively detecting all matching point pairs in a large dataset is intractable.

The benchmark is therefore built around comparing point pairs on surfaces that are guaranteed to be a correct match: two copies of the exact same surface. One of these two copies is left unmodified and represents the model, while one or more alterations are applied to the other copy to create a scene object. These alterations are applied as a sequence of one or more *filters*, where the output of one filter is used as the input for the next. Filters have fixed parameters, and are agnostic to any other filters applied on the sample object. After the scene mesh is computed, the effect on the matching capability of the descriptor can be measured by the distance between the descriptor pairs computed for corresponding points on the model and scene objects.

An overview over the benchmarking procedure is shown in Figure 2. A set of model objects is first drawn at random from a large dataset. For each of these models, 100 vertices are randomly selected from the object. The corresponding points on the scene mesh are located after the filter sequence has completed. Note that this may cause some points to be lost if the portion of the surface they were located on is removed by a filter. Each filter aims to simulate a real world phenomenon such as clutter (surfaces in the support volume that are not part of the model) and occlusion (portions of the object surface are missing due to these not being visible from the point of view of a 3D capturing device).

A descriptor pair is computed for each of the remaining point pairs, each respectively capturing corresponding points on the surface of the model and scene. This results in a model descriptor D_m, and a scene descriptor D_s that has undergone some modification. This descriptor pair is finally used to compute the Descriptor Distance Index and PRC/AUC metrics.

Each filter reports the value of the independent variable it simulates. This value may either be selected at random, or must be computed after the filter has completed. For example, a filter altering the orientation of normal vectors will report the rotation angle it randomly selected. A filter removing occluded surfaces can only compute the amount of area that was removed after it has been applied. The ability to vary such an independent variable comes from running the experiment many times on many different object pairs, increasing the likelihood that the variable happens to have a given value of interest. It is also worth noting that all independent variables must be computed on a point by point basis, rather than for the entire object. Figure 3 demonstrates why these values are location dependent.

3.1. The Descriptor Distance Index

The PRC was shown to be affected by the existence of multiple valid matches for the same surface. Accurately determining all these is computationally intractable. The nearest to second neighbour distance ratio σ is also affected by this issue, as two valid matches are likely to result in a high distance ratio. A secondary metric that is used in conjunction with the PRC, and avoids relying on measures that are susceptible to multiple similarity is therefore desirable. The Descriptor Distance Index (DDI) metric is therefore proposed.

In contrast to the PRC, the DDI aims to compare distances between corresponding model and scene point pairs directly. Unfortunately, this is not possible directly, as the computed distance values vary across descriptors and distance functions. An additional function is therefore needed to translate descriptor distances into a space that allows comparison.

Normalising all distances is not possible, as distance functions do not necessarily scale linearly. Using the same distance function for all methods risks disadvantaging some methods if another distance function would yield better performance. It is, however, possible to compare distances between a descriptor and other descriptors when using the same descriptor method and distance function.

The Descriptor Distance Index (DDI) thus relies on a large set of descriptors computed for random vertices sampled from randomly chosen objects from the dataset, called the *reference set R*. The DDI for a given pair of descriptors f_1, f_2 is defined as the cardinality of the set of descriptors from R that are closer in feature space to f_1 than f_2. The metric therefore effectively measures the extent to which f_2, from the perspective of f_1, is indistinguishable from noise. In our experiments, the size of the reference set was set to 1 000 000 descriptors.

The process for creating the reference set uses the same random seed for all of the tested methods. Each method is therefore asked to compute a descriptor for the exact same points from the exact same dataset objects, which ensures that all methods are tested on equal ground, and allows comparison of DDI values across methods. The metric is also not sensitive to the existence of multiple similar surfaces, as equivalent local surfaces should produce equivalent descriptors, and only descriptors whose distance value is *lower* are counted. If f_2 is computed over the same surface used to compute f_1 but has been altered in some way, the purpose of the metric still holds because f_2 is now ob-

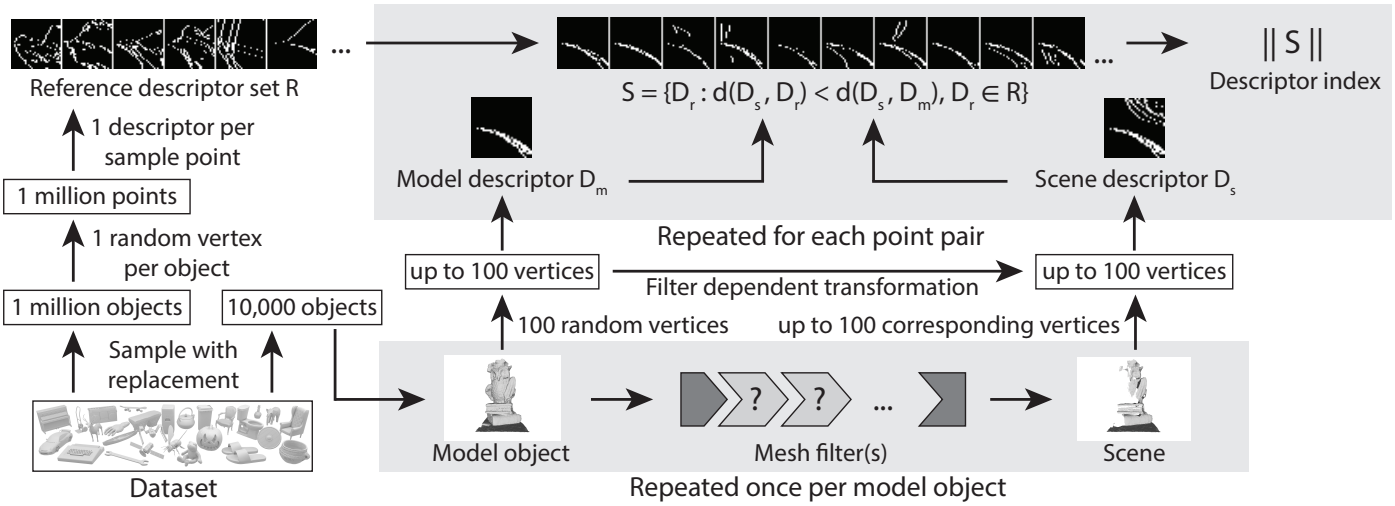


Fig. 2: An overview over the proposed benchmark.

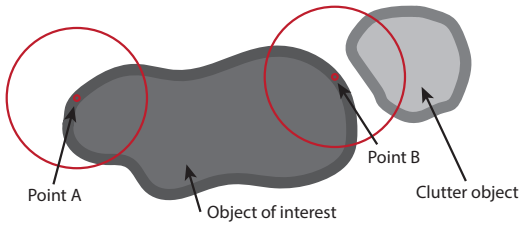


Fig. 3: A demonstration of a situation where a descriptor nuisance is localised to a portion on the object. Two descriptors are computed for two points and their support regions are shown. The support region of point A only contains the object of interest, while point B also contains clutter.

jectively less distinguishable from noise from the perspective of f_1.

3.2. Dataset

An appropriate dataset must be selected to serve as a model set and input to the matching conditions being tested by the filter sequence. The dataset should contain a wide variety of 3D data that is representative of the various use cases in which the tested methods may be applied.

The Objaverse dataset [25] was selected, which covers many domains such as household objects, furniture and vehicles. The dataset contains a total of 798,759 files from which 8,124 were excluded due to containing a point cloud or, in a few cases, for failing to parse. Point clouds were excluded because sampling triangle meshes into point clouds yields more similar surfaces across the two modalities compared to sampling point clouds into triangle meshes. The ABC dataset and ShapeNetCore datasets were also considered, but both mostly consist of CAD drawings with limited variety or application domains.

A derived version of the dataset was created to simplify distribution. This version only contains vertex positions, normals and, for 3D meshes, the polygon definitions for each object. The compression format is lossless and reduced the total size of the dataset from approximately 8.1 TB to approximately 1.5 TB.

3.3. Parameters

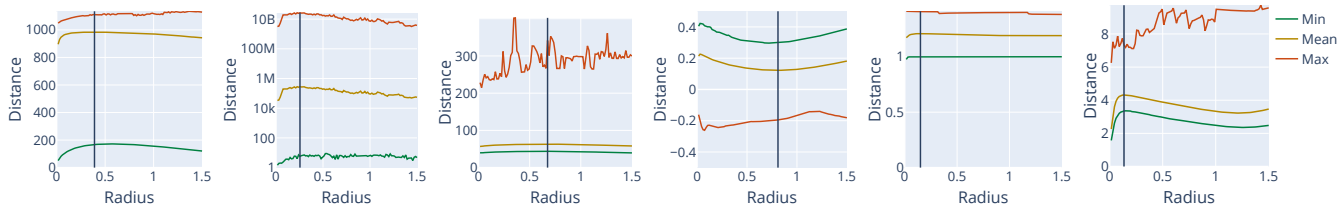
There are various parameters that must be selected in order to be able to compare matching performance in a manner that does not benefit specific methods. This includes the support radius, scale, and sample count used for sampling point clouds.

One downside of using artificial data is that there is no information available regarding the physical dimensions of each model. One option is to use the mesh resolution as an indication of scale, however this approach does not yield satisfactory scales for objects with a high variance in edge lengths. Objects are instead fitted into a unit sphere, which also aids the interpretation of any relevant distances in any produced results. This was achieved using the *seb* algorithm [44], and its publicly available implementation [45].

All local shape descriptors use a support volume surrounding the reference point to determine which surfaces to represent. A larger volume captures more surface information, but also has a greater risk of including clutter. Smaller volumes risk a reduction in descriptive capability. To date there is no commonly accepted or established approach to determining the support radius, and it is usually left up to the user to select.

From the perspective of fairness, a support radius determines how much surface information is given to the descriptor. It is inherently impossible to control the amount of information provided to each descriptor, as there exists variation in the shapes of the support volumes across methods. The strengths and weaknesses inherent to each method can affect how well a particular method performs in a benchmark. We thus conclude that a radius must be chosen on a per-method basis, and aim to choose the radius that maximises the method's capability to discriminate other non-matching descriptors.

The means by which this is achieved is to generate two sets of 100 000 descriptors for each support radius between 0.01 and 1.5 (recall that all objects are fitted into a unit sphere), with increments of 0.015. For each pair of sets, the average distance



(a) QUICCI (b) RICI (c) RoPS (d) Spin Image (e) SHOT (f) USC

Fig. 4: An overview over the lowest, mean, and highest distances observed across all 10^{10} descriptor pairs (all possible pairs from two sets of 100 000 descriptors each) for each support radius that was tested. The chosen support radius is indicated with a vertical line on each chart.

between all possible descriptor pairs in each set is computed. It is conjectured that the support radius that maximises this average distance would imply that the descriptor is on average optimally capable to discriminate its descriptors. This fixed support radius is subsequently used for all descriptors computed for that method. The set of models used for computing the support radius is different from the one used for selecting the set of models and reference descriptors.

A slightly modified version of the support radius selection procedure was used for the USC descriptor. The mechanism used by this descriptor to normalise bin contributions is not effective, and caused the average distances between descriptor pairs to decrease for higher support radii. The addition of a normalisation step corrected this problem. This step is only used during the support radius selection process.

Another relevant parameter is the number of point samples used to uniformly sample the triangle meshes from the dataset into point clouds. This step is needed when testing methods using these as input. The disparity between input modalities represents to some extent a source of unfairness between methods that use one or the other. A low resolution point cloud contains less information than the triangle mesh it was sampled from, while a high resolution greatly increases execution time.

Using the number of vertices or triangles, or mesh resolution of a mesh to set the sample count is not a good solution because the sizes of triangles can vary significantly, even within certain meshes. We therefore use a sample count of 1 000 000 per unit area. The area is calculated after fitting the object into a unit sphere. This ensures that all surfaces are sampled with a roughly equivalent resolution. To alleviate some of the effects of sampling noise and excessive computation time, a lower and upper bound of 1 000 000 and 5 000 000 points are used, respectively.

4. Filters

Each of the filters used for simulating various adverse matching conditions are now motivated and described in detail.

4.1. Clutter

This filter simulates the effects of clutter being present in the vicinity of the model, adding surfaces to the support volumes of tested descriptors that do not belong to the model itself. The intensity of clutter is measured using Equation 4.

$$Clutter = \frac{\text{Non model area in support volume}}{\text{Model area in support volume}} \tag{4}$$

The filter has been implemented by first sampling 10 clutter objects at random from the dataset. These added objects are subsequently simulated using the Jolt Physics library [46], which ensures objects adhere to physical constraints such as colliding with other objects and gravity. Objects are initially placed in a vertical stack in the air, after which gravity is applied and the objects fall on to a ground plane. The simulation ends when no more movement is detected. Clutter objects are attracted to the sample object to increase the likelihood that the objects form a pile.

One limitation of the Jolt Physics library is that it does not support the simulation of groups of arbitrary mesh surfaces. The V-HACD algorithm by Mammou et al. [47] was therefore used, through its publicly available implementation [48], to first subdivide each mesh into a set of convex hulls that approximate the original surface. These are used as a proxy during the simulation. This, in rare cases, yields degenerate hulls, which are removed.

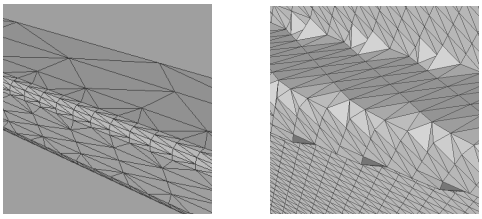
4.2. Occlusion

Occlusion is the result of surfaces not being visible from the point of view of a capture device, sometimes also referred to as *partiality*. This filter renders a high resolution image of the input scene from a random viewing direction, and removes any triangles that are not visible in the image. The intensity of the occlusion filter is given in Equation 5.

$$Occlusion = 1 - \frac{\text{Model area in scene in support volume}}{\text{Model area in support volume}} \tag{5}$$

4.3. Alternate mesh resolution

When an object is acquired using different acquisition methods, such as different 3D capture devices, the resolution of the produced mesh can vary due to variations in settings and hardware limitations. Testing variations in mesh resolution is a common occurrence in descriptor evaluations done to date. The prevalent way in which this is implemented in previous work is through the use of a decimation algorithm. The specific algorithms that are usually used for this purpose rely primarily on edge and half-edge collapse, along with a greedy scheme for deciding which edge to collapse next.



(a) (b)

Fig. 5: Two pictures of the same object surface where the original mesh in a has been captured by a simulated 3D capture device in b

While this achieves the desired effect of reducing mesh resolution, similar decimation rates can have an inconsistent effect on the mesh itself. A mesh consisting mostly of flat surfaces will see little physical change after reducing it to a low vertex count, while the same reduction applied to mesh with more organic shapes will be far more pronounced.

Another drawback of using decimation is that such algorithms often do not produce any of the sampling artefacts commonly found when using low quality capturing equipment. An example of this is shown in Figure 5, where the edge is represented unevenly due to sampling artefacts. A decimation algorithm would not be inclined to produce such meshes, as it attempts to maintain the shape of the mesh, and is more prone to simplify such ridges to sharp edges. Additionally, a low resolution scanner may be able to pick up smaller details, which a decimation algorithm is not guaranteed to keep. We therefore do not consider decimation to be a good resolution reduction strategy that is grounded in real world phenomena.

One potential solution that could be used instead is a remeshing algorithm, which attempts to recreate a mesh using approximately equilateral triangles with a given edge length. By using a target edge length that is larger than the average edge length of the original mesh, the resulting mesh should have fewer triangles than the original while approximating the original surface. However, this approach proved infeasible because a proper target edge length is difficult to establish. Specifying the target edge length either as a constant or using the average edge length of the input mesh risks creating an excessive number of triangles. This in turn causes high computation times and memory requirements in different implementations of remeshing algorithms that are currently available.

The adopted solution for this filter instead attempts to simulate an ideal low resolution scanner, by rendering the scene from a random point of view at a resolution of 640×480 pixels. The depth buffer is subsequently used to reconstruct the mesh. Variation in the mesh resolution is achieved by varying the distance of the object to the virtual depth camera. Due to the perspective projection, the object will on average cover fewer pixels in the depth buffer, thus being reconstructed using fewer triangles.

4.4. Alternate triangulation

When the same object is captured repeatedly, the produced mesh surface should be similar in shape when assuming the capture quality was reasonable. However, the manner in which the surface is triangulated is unlikely to be similar due to various sources of noise during the reconstruction process. An example of this is shown in Figure 6. If a method should thus be able to recognise an equivalent surface, it must be capable of doing so irrespective of how that surface is represented.

Furthermore, even if a keypoint detector is able to locate the same keypoint in both mesh variants, the exact location of each keypoint relative to the original surface may have shifted.

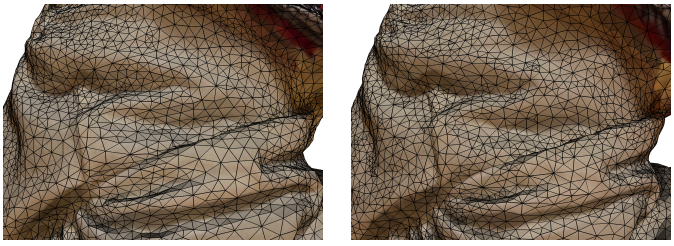


Fig. 6: Two different 3D captures of the same object. The surfaces being represented are the same, but the positions of vertices and triangles is different between them.

Remeshing is a good candidate for implementing a similar effect to the alternate mesh resolution filter. However, as stated in Section 4.3, current remeshing algorithms were not found to be viable. We instead used a mesh smoothing algorithm proposed by Surazhsky and Gotsman [49], and its implementation from the CGAL library [50]. The algorithm adjusts vertex positions to form higher quality triangles (e.g. more equilateral in shape, and similar in area), while maintaining the overall shape of the mesh. The result is a similar mesh with displaced vertices, which is in line with the objective of this filter. The intensity of the effect of this filter is measured by computing the distance to nearest vertex on the filtered mesh for each point on the model surface.

4.5. Deviated normal vector

Many methods for estimating normal vectors have been proposed to date. However, factors such as noisy input data and estimation errors can propagate to deviations in the computed normal vectors. Understanding how these deviations affect the matching performance of a method is therefore relevant. This filter adjusts all normals of the input model by computing a new normal vector that deviates from the original by a uniformly sampled random angle. The selected angle is chosen to be between 0 and 30 degrees. The azimuth direction in which the normal is rotated is also chosen randomly.

4.6. Deviated support radius

Calibration or estimation errors in 3D capturing equipment can cause the scale of a produced mesh to vary slightly across repeated captures. Alternatively, if the support radius for a point is selected using an algorithm on a per point basis, errors in the radius estimation may cause a similar effect.

The filter scales the support radius by a randomly chosen factor s between 0.75 and 1.25. For the ease of implementation, this is done by scaling the mesh by a factor of $2 - s$, which achieves the same effect.

4.7. Gaussian noise

Noise is a common occurrence in captured 3D data due to various sources of inaccuracies during the capturing process. Each vertex with a unique vertex position is displaced by a distance sampled from a normal distribution. The displacement direction is chosen by computing the average direction of the normal vectors of all vertices sharing the same vertex position. The same standard deviation is used for all vertices in the object, whose value is selected randomly between 0.0001 and 0.01. These were chosen to be reasonable perturbations for what can be expected of scans of varying quality.

5. Results

The proposed evaluation methodology is used to evaluate the QUICCI, RICI, USC, Spin Image, SHOT, and RoPS descriptors. These were chosen to be a representative set of both popular classic descriptors, while also including some that have been proposed more recently. The used support radius, distance functions, and other method specific parameters are listed in Table 2. The AUC_{pr} was computed using the set of all model descriptors, in accordance with its implementation in previous work.

The USC method should be noted specifically here. The input point clouds for this particular method were downsampled to 1% of the number of points used for other point cloud based methods. The descriptor requires the computation of a *point density* value for each point in the point cloud, which is an $\mathrm{O}(n^2)$ operation. This was done because computing descriptors at the full point cloud resolution proved intractable. The results in this section for this method show that this has likely influenced the matching performance of this method, as it is not consistently able to identify identical geometry. We consider the results for this method valid despite this problem, because using this method in a practical context would likely be done using a downsampled point cloud anyway. However, it is likely that the matching performance for the full resolution point clouds would be higher.

The benchmark is run for a total of 10 filter configurations, one for each of the 7 presented filters where that filter is run by itself, and three combinations of two filters. The same root random seed is used for all of these configurations, which means that the reference set R, the set of sample objects, and the surface points sampled from those objects are all identical across all iterations of the benchmark process.

Each experiment produces 1 000 000 data points, although depending on the filter(s) that are applied, a portion of these may be lost. For example, the occlusion filter removes any vertices that are not visible from the perspective of the camera. A plot with the distribution of sample counts is therefore given alongside the observed matching performance for each filter configuration.

The benchmark itself was implemented in C++, using the descriptor implementations from the libShapeDescriptor library [52]. While the library contains GPU implementations for a number of the used descriptors, we found that using the CPU variants was more effective for the purposes of this benchmark, as the quantity of descriptors being computed at a time was not sufficient to saturate the stream processors of the GPU with work, making that path slower than using the CPU. However, our implementation of the benchmark does support descriptors implemented as GPU kernels.

5.1. Single filter experiments

The charts in this section measure the effect of a single filter (and as such a single independent variable). Each chart contains a visualisation of the distribution of DDI values, as well as a curve showing the computed AUC_{pr} for the same set of results. Values of the DDI can vary between 0 and the size of the reference descriptor set, which has been set to 1 000 000. The charts are constructed by first dividing the range of the x-axis into 75 subdivisions. For all sample points whose x-coordinates fall in each of these subdivisions, the share of DDI values that falls into each order of magnitude is computed. The number of samples in each order of magnitude is subsequently normalised to the total number of point samples in that subdivision, which yields the *proportional DDI* value. The AUC_{pr} value computed for each subdivision is plotted alongside these.

A higher proportion of low DDI values is desirable. Ideally, all DDI values are zero, which would result in their relative proportion being 1 for each subdivision. An example where this is almost the case can be seen in Figure 10d. A DDI value of 0 indicates that the method uniquely identified the correct model descriptor out of all descriptors in the reference set.

The higher ranges of DDI values visualise how quickly matching performance deteriorates, which provides more context than a single curve. An example of this can be seen in Figure 11a and 11b, where only plotting the proportion of samples that have a DDI of 0 would have shown both methods to be approximately equivalent in performance, while the proportional DDI shows a more rapid decline in performance in the case of the RICI descriptor. This is shown as a larger proportion of higher DDI values.

The plots in this section indicate that values of AUC_{pr} and the fraction of results whose DDI value was measured to be 0 are often similar. There are some exceptions, such as Figures 8a, 8b, and 12c. Limited testing indicates that these discrepancies are primarily caused by a poor σ ratio. However, excluding the σ ratio still left a similar gap to those seen on other plots. It may be possible to explain this discrepancy by the existence of multiple viable matches. Its impact thus appears to be measurable, but limited.

5.1.1. Clutter

Figure 7 shows the results of the experiment where solely the clutter filter is applied on the model object. The RICI descriptor is shown to be highly resistant to the effects of clutter here. while USC, SHOT, and RoPS show poor resistance. We conjecture that RoPS is rather sensitive to clutter due to the histogram step using a bounding box that covers all point samples present in the support volume. When clutter is added, the dimensions of this bounding box change, causing the histogram to lose correspondence with its clutter free counterpart. RoPS and SHOT both also include a normalisation step, which may be sensitive to the presence of clutter.

Method	Support radius	Method specific parameters	Distance Function
QUICCI	0.39	Resolution: 31×32	Weighted Hamming [51]
RICI	0.255	Resolution: 32×32	Clutter resistant squared sum of differences [33]
RoPS	0.675	Point samples per unit area: 100 000 Point sample limit: 5 000 000	Euclidean distance
Spin Image	0.81	Resolution: 32×32, support angle A_s: 180°	Pearson correlation
USC	0.135	Resolution: J = 10, K = 14, L = 14 r_{min}: 0.014, δ: 0.01, sampling density: 1%	Euclidean distance
SHOT	0.15	Resolution: $s = 11$, $\lambda = 8$, $\mu = 2$, $R = 2$	Euclidean distance

Table 2: An overview over the tested methods and the relevant parameters used.

In the case of USC, despite the contributions of individual points being normalised by the local point density when the histogram is constructed, the effect of clutter is that the values of individual descriptor bins are increased. This in turn results in added distance to its nearest neighbour due to the use of the Euclidean distance function. Clutter has a similar effect on the Spin Image, but the use of the Pearson Correlation distance function likely reduces some of the impact.

The sample counts in Figure 7g show how the distribution of clutter varies across different support radii. The methods with larger support radii experience larger amounts of clutter more often, as would be expected.

5.1.2. Occlusion

The results for the experiment where only the occlusion filter was applied are shown in Figure 8. Here the Spin Image performs best. The QUICCI and RICI descriptors demonstrate a capability of correctly identifying the model descriptor when portions of the object surface are missing. However, as discussed previously, this comes at the cost of lower σ ratios. These may partially be explained in the case of the QUICCI descriptor by a reduction in the number of set bits (to 1, specifically) by the occlusion filter. The remaining set bits are more likely to better overlap with more distant neighbours, which is emphasised by the used weighted Hamming distance function.

The sample count distribution shows that the occurrence of a partiality of 50% is common, despite the variation in support radii amongst the tested methods. With respect to replicability, there are small variations induced into the results, depending on which OpenGL implementation is used. We have used the one provided by Mesa 23.1.4. The same applies to the alternate mesh resolution filter.

We compared our results for the clutter and occlusion filters to those presented by Guo et al. [5] in Fig. 7g and 7h. Both Figures appear to exhibit high levels of noise, where most curves fluctuate to varying degrees. In contrast, our quantitative results, computed over approximately two to three orders of magnitude more sample points, indicate that these curves should in most cases be monotonically decreasing with increasing levels of clutter and occlusion. While the authors were not able to determine one themselves, it appears that a random error is likely present in the data.

For the USC descriptor, neither of the reported curves matches with the conclusions of our evaluation. The Spin image results for occlusion show some similarity, where most observations are within an estimated error margin of 0.2 AUC_{pr}. The reported occlusion results for RoPS are in line with our own results.

5.1.3. Alternate triangulation

For the alternate triangulation filter, whose results are shown in Figure 9, only weak correlation was observed between the average edge length (mesh resolution), and the matching performance of the different descriptors. Of the tested descriptors, RoPS and SHOT exhibit similar matching performance, with USC performing best. Our testing did not show a relationship between the matching performance of a descriptor, and the vertex count of the input mesh.

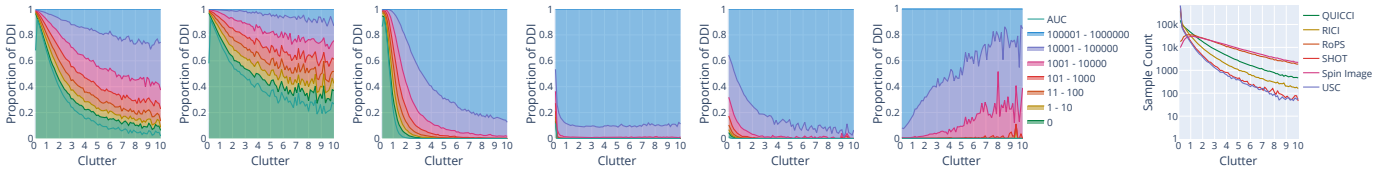
5.1.4. Deviated normal vector

The deviated normal vector filter randomly chooses the angle by which the normal vector of the scene mesh point is perturbed. This yields an even distribution of point samples across the different rotation angles. The QUICCI, RICI, SHOT, and Spin Image descriptors use the normal vector of a keypoint to orient their histograms, and are affected by perturbations to this vector, as can be seen in Figure 10.

The QUICCI and RICI descriptors rely on the similarity of rasterised local contours (e.g. the circular shape of a bicycle wheel). A rotation of the normal vector would cause the position of these rasterised contours to shift within the descriptor image. We conjecture that this is the cause of the drop in matching performance when the normal vector deviation angle is increased.

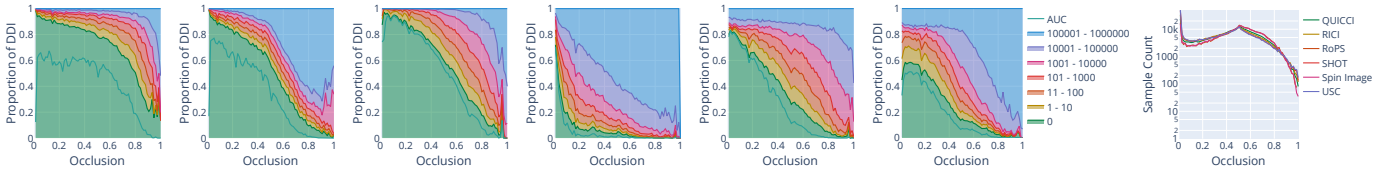
A similar effect occurs in the case of the Spin Image, which, instead of intersection counts, estimates the mesh surface area intersecting a histogram bin, when that bin is rotated around a common axis for one rotation [53]. Its improved performance over the QUICCI and RICI descriptors may be explained by that changes in the area intersecting with each bin with increasing normal vector deviation angles are more gradual than intersection counts.

The SHOT descriptor performs better at higher normal vector deviation angles than the QUICCI, RICI, and Spin Image descriptors. We conjecture that this is caused by the comparatively large volume described by each histogram bin. The SHOT descriptor does not achieve perfect matching performance when the normal vector is left intact. This can be explained by that the filter also modifies the normals of all vertices in the scene. The SHOT descriptor uses these to compute its histograms.



(a) QUICCI (b) RICI (c) Spin Image (d) RoPS (e) SHOT (f) USC (g) Sample distribution

Fig. 7: The effect of varying levels of clutter on the matching performance of various descriptors. Figure 7g shows the number of sample points per histogram bin.



(a) QUICCI (b) RICI (c) Spin Image (d) RoPS (e) SHOT (f) USC (g) Sample distribution

Fig. 8: The effect of varying levels of occlusion on the matching performance of various descriptors. Figure 8g shows the number of sample points per histogram bin.

The USC and RoPS descriptors do not utilise the information of normal vectors, but their results have been included because it shows that RoPS achieves a near perfect matching score when provided with effectively equivalent geometry. The same is true for USC, although as mentioned previously, the low resolution used for computing these descriptors appears to diminish its matching capabilities.

5.1.5. *Deviated support radius*

The results of the support radius deviation filter in Figure 11 show that the different descriptors have varying sensitivity levels for scale and/or support radius mismatches. The performance of QUICCI and RICI can be explained with reasons that are similar to those outlined for the deviated normal vector. The rasterisation done by these descriptors relies on intersection counts occurring at specific distances, and when these are displaced by a change in scale, the observed matching performance drops. QUICCI appears to be slightly more resistant than RICI. The SHOT descriptor demonstrates excellent performance in this filter. The filter chooses the applied scale factor from a uniform distribution, which thus results in an approximately constant sample distribution.

5.1.6. *Gaussian noise*

When applying varying levels of Gaussian noise, the results in Figure 12 show that the Spin Image and SHOT descriptor are highly resistant. This may for the Spin Image be explained by that the area (by proxy the number of sample points) does not change much with higher levels of noise.

In the case of the QUICCI and RICI descriptor, the roughness of the surface induces additional variations in number of intersection counts observed by the descriptor, reducing its ability to discriminate. For the USC descriptor, only 900 000 results were computed due to the descriptor experiencing excessive execution times.

5.1.7. *Alternate mesh resolution*

The final single filter experiment is applying the alternate mesh resolution filter, whose results are shown in Figure 13. For this filter, the centre of the sample object is placed at a randomly selected distance from the camera. While the matching performance for all methods is poor, the Spin Image appears to be most resistant to the reduced reconstructed mesh resolution.

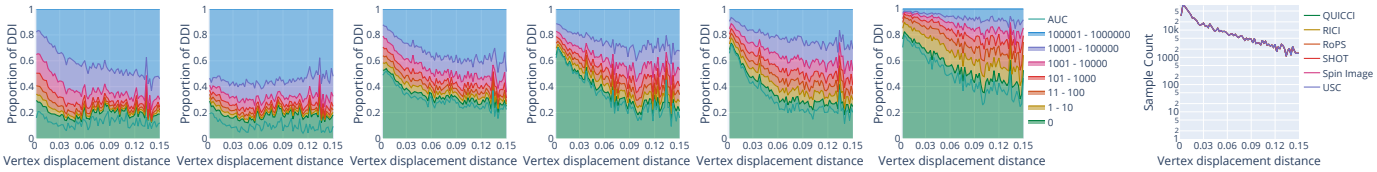
5.1.8. *Summary*

In order to gain an overview over how well each method performs across the different filters, we computed a summary chart, shown in Figure 14. We used a similar approach to the PRC evaluation methodology, by computing the area underneath the curve where DDI is 0. It should be noted that while performance can be compared across methods within the same filter, it does not directly translate between different filters, due to each filter imposing different matching conditions on the scene. The range of each independent variable was also selected arbitrarily for each chart, and the area under the DDI curve represents the extent to which a method has achieved good performance in the entirety of that range. The intent of the chart is to highlight cases where a method might perform better or worse relative to the other tested methods.

The chart shows that the QUICCI and RICI descriptors are clearly superior for cluttered environments. The performance of the RoPS, SHOT, and USC descriptors in cluttered scenes are not missing in this plot. The area described by their DDI curves is small. QUICCI and RICI also perform well in occluded scenes –along with the Spin Image– but exhibit comparatively weak performance when confronted with various types of noise, where the Spin Image, SHOT, and USC descriptors excel. The only exception is deviations in the normal vector, to which the Spin Image is sensitive.

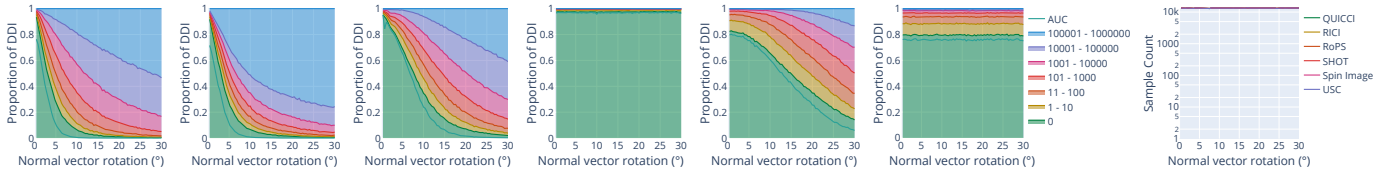
5.2. *Dual filter experiments*

The dual filter experiments use a pipeline with two filters each, and are thus capturing the effects of two independent variables. To visualise these, a 2D heatmap is used, which counts



(a) QUICCI (b) RICI (c) Spin Image (d) RoPS (e) SHOT (f) USC (g) Sample distribution

Fig. 9: Results for the alternate triangulation filter. Figure 9g shows the number of sample points per histogram bin.



(a) QUICCI (b) RICI (c) Spin Image (d) RoPS (e) SHOT (f) USC (g) Sample distribution

Fig. 10: Results for the deviated normal vector filter. The horizontal axis represents the angle by which the normal vector of the scene point was rotated. Figure 10g shows the number of sample points per histogram bin.

the fraction of samples whose DDI is 0. If a bin has less than 5 samples, it is removed. Removed bins show a background grid instead of a heatmap pixel.

The first of the tested dual filter pipelines is the clutter filter followed by the occlusion filter, which is a common occurrence in physical environments. The results of this experiment are shown in Figure 15. We observe here that higher levels of clutter have a self-occluding effect, limiting the observed level of clutter in the final scene mesh.

For the configurations for which data is available, the USC, RoPS, and SHOT descriptors show a poor capability of handling any combination of occlusion and clutter. RICI appears to perform best.

The final two configurations combine a clutter and occlusion filter, with a Gaussian noise filter, respectively. The results for the pipeline containing the clutter filter are shown in Figure 16, and the results for the pipeline containing the occlusion filter in Figure 17.

For the pipeline with the clutter filter, RICI shows the highest performance across the tested methods, and maintains much of this performance when higher intensities of Gaussian noise are applied. In the case of the pipeline with the occlusion filter, the Spin Image performs best out of of the tested descriptors.

6. Conclusion

The ShapeBench benchmark has been proposed, along with the novel DDI metric as an extension to the popular AUC_{pr} metric. The DDI metric was shown to be more informative than solely using the area under precision-recall curves, while simultaneously being insensitive to multiple occurrences. The combination of the AUC_{pr} and DDI yield an improved insight in the expected performance of a local 3D shape descriptor than each does individually.

The benchmark was used to test a range of descriptor methods, demonstrating their strengths and weaknesses under various conditions. The replicability of the produced results, and the public availability of the source code may both assist in gaining a deeper understanding of previous work, as well as the development and evaluation of new descriptor methods in the future.

Our results show that the number of models and point samples used in the evaluations of previous work may not have achieved a precision adequate to compare methods.

6.1. Future Work

While the presented benchmark includes a strategy for automatic selection of support radii, it is not necessarily optimal. Human environments contain many shapes at varying scales. For instance, a building viewed from the outside may be roughly shaped as a cuboid, while on the inside contains smaller shapes such as chairs and tables. It is possible to think of a support radius as an indication of the expected scale of shapes within, and algorithms estimating such scale should be investigated. It may even prove necessary to create multiple descriptors per keypoint.

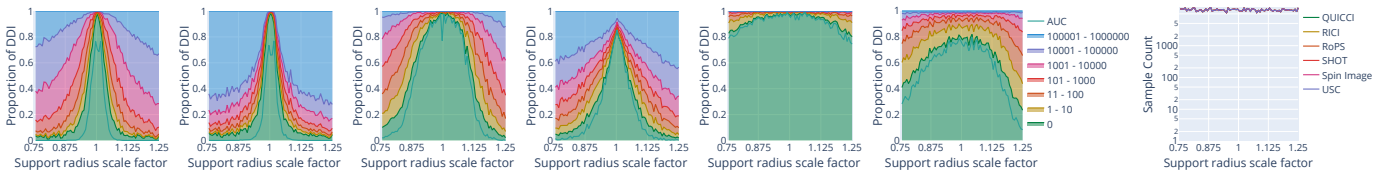
The benchmark has also not investigated the effect of different distance functions on the matching performance of different descriptors.

Acknowledgments

I would like to thank Jonathan Brooks and Haakon Gunnarsli for our interesting discussions, Theoharis Theoharis for his generous hardware donations to this project, and Ingulf Helland for providing additional compute resources. A total of 156 890 CPU hours were provided by the IDUN cluster [54].

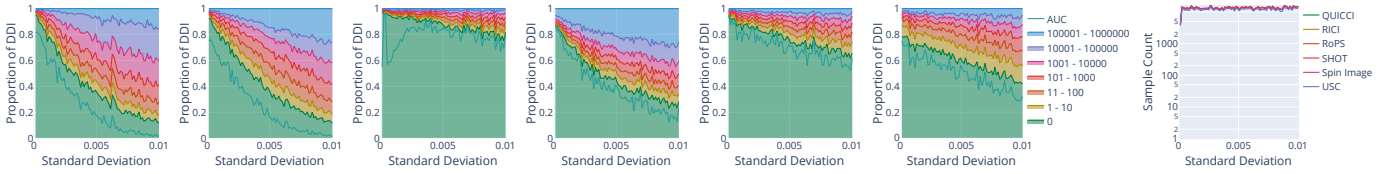
References

[1] Tam, GK, Cheng, ZQ, Lai, YK, Langbein, F, Liu, Y, Marshall, AD, et al. Registration of 3D Point Clouds and Meshes: A Survey from Rigid to Nonrigid. IEEE Transactions on Visualization and Computer Graphics 2013;19(7):1199–1217. URL: https://doi.org/10.1109/TVCG.2012.310. doi:10.1109/TVCG.2012.310.



(a) QUICCI (b) RICI (c) Spin Image (d) RoPS (e) SHOT (f) USC (g) Sample distribution

Fig. 11: Results for the support radius deviation filter. Figure 11g shows the number of sample points per histogram bin.



(a) QUICCI (b) RICI (c) Spin Image (d) RoPS (e) SHOT (f) USC (g) Sample distribution

Fig. 12: Results the Gaussian noise filter. Figure 12g shows the number of sample points per histogram bin.

[2] Kumbla, NB, Thakar, S, Kaipa, KN, Marvel, J, Gupta, SK. Handling Perception Uncertainty in Simulation-Based Singulation Planning for Robotic Bin Picking. Journal of Computing and Information Science in Engineering 2018;18(2). URL: https://doi.org/10.1115/1.4038954. doi:10.1115/1.4038954.

[3] Cui, Y, Chen, X, Zhang, Y, Dong, J, Wu, Q, Zhu, F. BoW3D: Bag of Words for Real-Time Loop Closing in 3D LiDAR SLAM. IEEE Robotics and Automation Letters 2023;8(5):2828–2835. URL: https://ieeexplore.ieee.org/abstract/document/9944848. doi:10.1109/LRA.2022.3221336; conference Name: IEEE Robotics and Automation Letters.

[4] Savva, M, Yu, F, Su, H, Kanezaki, A, Furuya, T, Ohbuchi, R, et al. Large-Scale 3D Shape Retrieval from ShapeNet Core55. The Eurographics Association; 2017. ISBN 978-3-03868-030-7. URL: https://diglib.eg.org:443/xmlui/handle/10.2312/3dor20171050. doi:10.2312/3dor.20171050; accepted: 2017-04-22T17:17:41Z ISSN: 1997-0471.

[5] Guo, Y, Bennamoun, M, Sohel, F, Lu, M, Wan, J, Kwok, NM. A Comprehensive Performance Evaluation of 3D Local Feature Descriptors. International Journal of Computer Vision 2016;116(1):66–89. URL: https://doi.org/10.1007/s11263-015-0824-y. doi:10.1007/s11263-015-0824-y.

[6] Zhao, B, Chen, X, Le, X, Xi, J. A quantitative evaluation of comprehensive 3D local descriptors generated with spatial and geometrical features. Computer Vision and Image Understanding 2020;190:102842. URL: https://www.sciencedirect.com/science/article/pii/S1077314218302698. doi:10.1016/j.cviu.2019.102842.

[7] Huang, H, Kalogerakis, E, Chaudhuri, S, Ceylan, D, Kim, VG, Yumer, E. Learning Local Shape Descriptors from Part Correspondences with Multiview Convolutional Networks. ACM Transactions on Graphics 2017;37(1):6:1–6:14. URL: https://dl.acm.org/doi/10.1145/3137609. doi:10.1145/3137609.

[8] Zeng, A, Song, S, Nießner, M, Fisher, M, Xiao, J, Funkhouser, T. 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017, p. 199–208. URL: https://ieeexplore.ieee.org/document/8099512. doi:10.1109/CVPR.2017.29; iSSN: 1063-6919.

[9] Hao, L, Yang, X, Xu, K, Yi, W, Shen, Y, Wang, H. Rotational Voxels Statistics Histogram for both real-valued and binary feature representations of 3D local shape. Journal of Visual Communication and Image Representation 2023;93:103817. URL: https://www.sciencedirect.com/science/article/pii/S1047320323000676. doi:10.1016/j.jvcir.2023.103817.

[10] Bibissi, DL, Yang, J, Quan, S, Zhang, Y. Dual spin-image: A bi-directional spin-image variant using multi-scale radii for 3D local shape description. Computers & Graphics 2022;103:180–191. URL: https://www.sciencedirect.com/science/article/pii/S0097849322000310. doi:10.1016/j.cag.2022.02.010.

[11] Tao, W, Hua, X, Yu, K, Chen, X, Zhao, B. A Pipeline for 3-D Object Recognition Based on Local Shape Description in Cluttered Scenes. IEEE Transactions on Geoscience and Remote Sensing 2021;59(1):801–816. URL: https://ieeexplore.ieee.org/abstract/document/9115889. doi:10.1109/TGRS.2020.2998683; conference Name: IEEE Transactions on Geoscience and Remote Sensing.

[12] Zhao, H, Tang, M, Ding, H. HoPPF: A novel local surface descriptor for 3D object recognition. Pattern Recognition 2020;103:107272. URL: https://www.sciencedirect.com/science/article/pii/S0031320320300777. doi:10.1016/j.patcog.2020.107272.

[13] Quan, S, Ma, J. On shortened 3D local binary descriptors. Information Sciences 2020;510:33–49. URL: https://www.sciencedirect.com/science/article/pii/S0020025519308795. doi:10.1016/j.ins.2019.09.028.

[14] Yang, J, Zhang, Q, Xian, K, Xiao, Y, Cao, Z. Rotational contour signatures for both real-valued and binary feature representations of 3D local shape. Computer Vision and Image Understanding 2017;160:133–147. URL: https://www.sciencedirect.com/science/article/pii/S1077314217300322. doi:10.1016/j.cviu.2017.02.004.

[15] Tombari, F, Salti, S, Di Stefano, L. Performance Evaluation of 3D Keypoint Detectors. International Journal of Computer Vision 2013;102(1):198–220. URL: https://doi.org/10.1007/s11263-012-0545-4. doi:10.1007/s11263-012-0545-4.

[16] Tombari, F, Salti, S, Di Stefano, L. Unique shape context for 3d data description. In: Proceedings of the ACM workshop on 3D object retrieval - 3DOR '10. Firenze, Italy: ACM Press. ISBN 978-1-4503-0160-2; 2010, p. 57. URL: http://portal.acm.org/citation.cfm?doid=1877808.1877821. doi:10.1145/1877808.1877821.

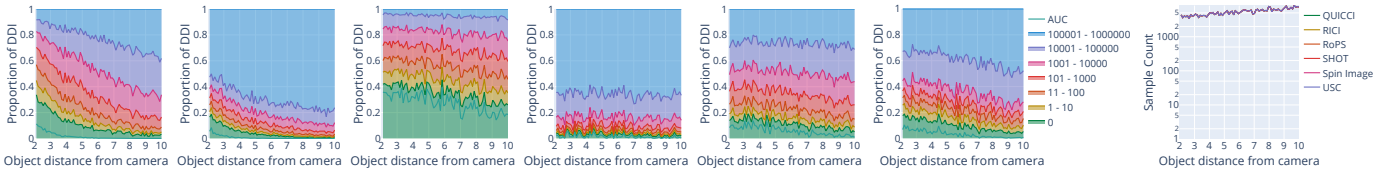
[17] Mian, A, Bennamoun, M, Owens, R. Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence 2006;28(10):1584–1601. URL: https://ieeexplore.ieee.org/document/1677516. doi:10.1109/TPAMI.2006.213; conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[18] Mian, A, Bennamoun, M, Owens, R. On the Repeatability and Quality of Keypoints for Local Feature-based 3D Object Retrieval from Cluttered Scenes. International Journal of Computer Vision 2010;89(2):348–361. URL: https://doi.org/10.1007/s11263-009-0296-z. doi:10.1007/s11263-009-0296-z.

[19] Salti, S, Tombari, F, Di Stefano, L. SHOT: Unique signatures of histograms for surface and texture description. Computer Vision and Image Understanding 2014;125:251–264. URL: https://www.sciencedirect.com/science/article/pii/S1077314214000988. doi:10.1016/j.cviu.2014.04.011.

[20] Taati, B, Bondy, M, Jasiobedzki, P, Greenspan, M. Variable Dimensional Local Shape Descriptors for Object Recognition in Range Data. In: 2007 IEEE 11th International Conference on Computer Vision. 2007, p. 1–8. doi:10.1109/ICCV.2007.4408830; iSSN: 2380-7504.

[21] Shotton, J, Glocker, B, Zach, C, Izadi, S, Criminisi, A, Fitzgibbon, A.



(a) QUICCI (b) RICI (c) Spin Image (d) RoPS (e) SHOT (f) USC (g) Sample distribution

Fig. 13: Results the alternate mesh resolution filter. Figure 13g shows the number of sample points per histogram bin.

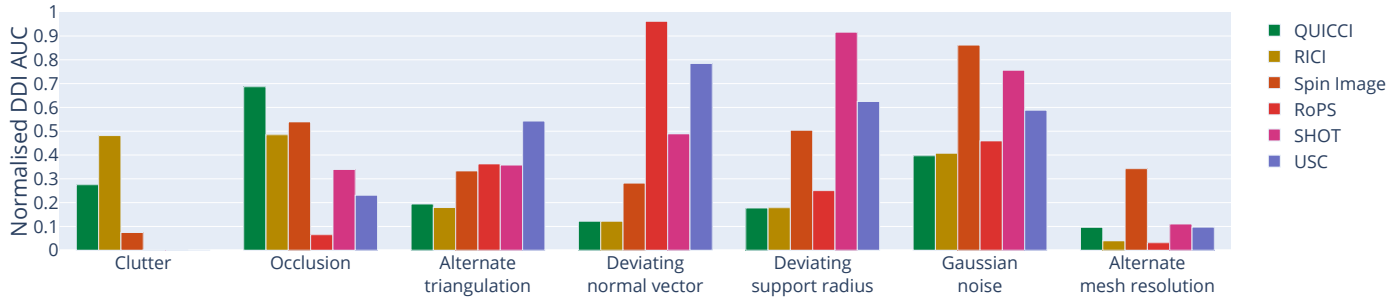
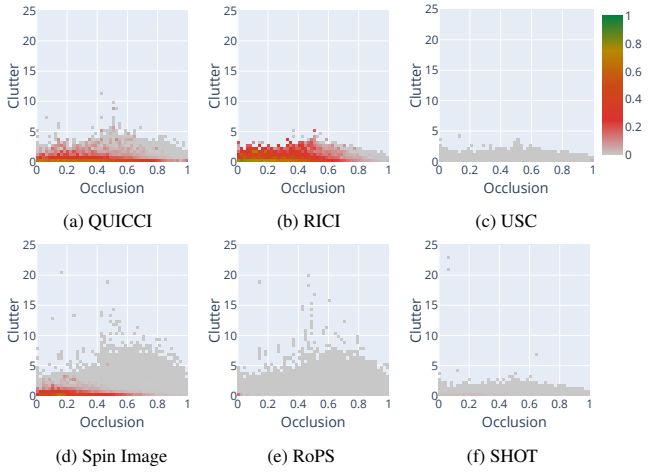
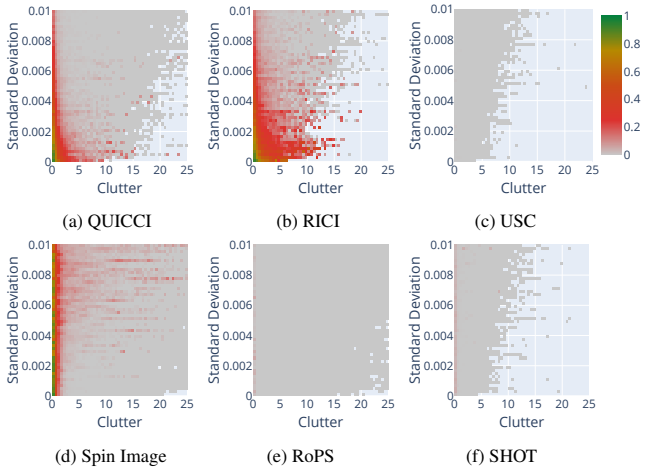


Fig. 14: An overview over the relative performance of the tested methods across each of the filters. Performance is measured as the normalised area under the curve where the proportional DDI is zero.



(a) QUICCI (b) RICI (c) USC (d) Spin Image (e) RoPS (f) SHOT

Fig. 15: Results for the clutter filter followed by the occlusion filter.



(a) QUICCI (b) RICI (c) USC (d) Spin Image (e) RoPS (f) SHOT

Fig. 16: Results for the clutter filter followed by the Gaussian noise filter.

Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition. 2013, p. 2930–2937. URL: https://ieeexplore.ieee.org/document/6619221. doi:10.1109/CVPR.2013.377; iSSN: 1063-6919.

[22] Sølund, T, Buch, AG, Krüger, N, Aanæs, H. A Large-Scale 3D Object Recognition Dataset. In: 2016 Fourth International Conference on 3D Vision (3DV). 2016, p. 73–82. doi:10.1109/3DV.2016.16.

[23] Chang, AX, Funkhouser, T, Guibas, L, Hanrahan, P, Huang, Q, Li, Z, et al. ShapeNet: An Information-Rich 3D Model Repository. 2015. URL: http://arxiv.org/abs/1512.03012. doi:10.48550/arXiv.1512.03012; arXiv:1512.03012 [cs].

[24] Koch, S, Matveev, A, Jiang, Z, Williams, F, Artemov, A, Burnaev, E, et al. ABC: A Big CAD Model Dataset for Geometric Deep Learning. 2019, p. 9601–9611. URL: https://openaccess.thecvf.com/content_CVPR_2019/html/Koch_ABC_A_Big_CAD_Model_Dataset_for_Geometric_Deep_Learning_CVPR_2019_paper.html.

[25] Deitke, M, Schwenk, D, Salvador, J, Weihs, L, Michel, O, VanderBilt, E, et al. Objaverse: A universe of annotated 3D objects. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023, p. 13142–13153.

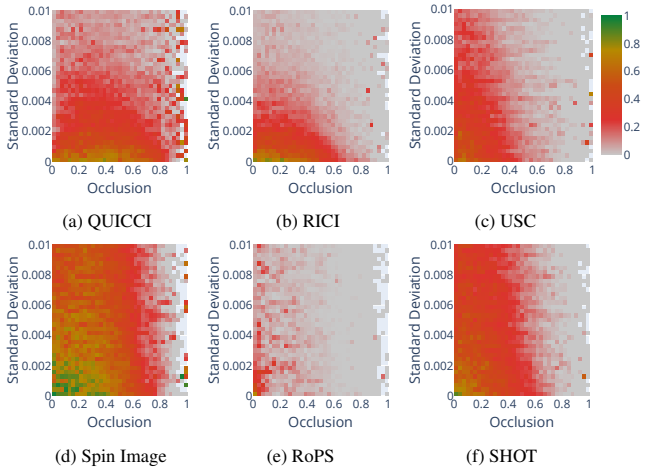
[26] Lowe, DG. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 2004;60(2):91–110. URL: https://doi.org/10.1023/B:VISI.0000029664.99615.94. doi:10.1023/B:VISI.0000029664.99615.94.

[27] Flint, A, Dick, A, Hengel, Avd. Local 3D structure recognition in range images. IET Computer Vision 2008;2(4):208–217. URL: https://digital-library.theiet.org/content/journals/10.1049/iet-cvi_20080037. doi:10.1049/iet-cvi:20080037; publisher: IET Digital Library.

[28] Darom, T, Keller, Y. Scale-Invariant Features for 3-D Mesh Models. IEEE Transactions on Image Processing 2012;21(5):2758–2769. doi:10.1109/TIP.2012.2183142; conference Name: IEEE Transactions on Image Processing.

[29] Osada, R, Funkhouser, T, Chazelle, B, Dobkin, D. Shape distributions. ACM Transactions on Graphics 2002;21(4):807–832. URL: https://doi.org/10.1145/571647.571648. doi:10.1145/571647.571648.

[30] Wahl, E, Hillenbrand, U, Hirzinger, G. Surflet-pair-relation histograms: a statistical 3D-shape representation for rapid classification. In: Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. 2003, p. 474–481. doi:10.1109/IM.2003.



(a) QUICCI (b) RICI (c) USC

(d) Spin Image (e) RoPS (f) SHOT

Fig. 17: Results for the occlusion filter followed by the Gaussian noise filter.

1240284.

[31] Hanley, JA, McNeil, BJ. The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 1982;143(1):29–36. URL: https://pubs.rsna.org/doi/10.1148/radiology.143.1.7063747. doi:10.1148/radiology.143.1.7063747; publisher: Radiological Society of North America.

[32] Kim, VG, Li, W, Mitra, NJ, Chaudhuri, S, DiVerdi, S, Funkhouser, T. Learning part-based templates from large collections of 3D shapes. ACM Transactions on Graphics 2013;32(4):70:1–70:12. URL: https://dl.acm.org/doi/10.1145/2461912.2461933. doi:10.1145/2461912.2461933.

[33] van Blokland, BI, Theoharis, T. Radial intersection count image: A clutter resistant 3D shape descriptor. Computers & Graphics 2020;91:118–128. URL: https://www.sciencedirect.com/science/article/pii/S0097849320301096. doi:10.1016/j.cag.2020.07.007.

[34] Blokland, BIv, Theoharis, T. An indexing scheme and descriptor for 3D object retrieval based on local shape querying. Computers & Graphics 2020;92:55–66. URL: https://www.sciencedirect.com/science/article/pii/S009784932030128X. doi:10.1016/j.cag.2020.09.001.

[35] Buch, AG, Petersen, HG, Krüger, N. Local shape feature fusion for improved matching, pose estimation and 3D object recognition. SpringerPlus 2016;5(1):297. URL: https://doi.org/10.1186/s40064-016-1906-1. doi:10.1186/s40064-016-1906-1.

[36] Laga, H, Guo, Y, Tabia, H, Fisher, RB, Bennamoun, M. 3D Shape Analysis: Fundamentals, Theory, and Applications. John Wiley & Sons; 2018. ISBN 978-1-119-40518-4. Google-Books-ID: zzKBDwAAQBAJ.

[37] Gao, Y, Dai, Q, Zhang, NY. 3D model comparison using spatial structure circular descriptor. Pattern Recognition 2010;43(3):1142–1151. URL: https://www.sciencedirect.com/science/article/pii/S0031320309002933. doi:10.1016/j.patcog.2009.07.012.

[38] Papadakis, P, Pratikakis, I, Theoharis, T, Perantonis, S. PANORAMA: A 3D Shape Descriptor Based on Panoramic Views for Unsupervised 3D Object Retrieval. International Journal of Computer Vision 2010;89(2-3):177–192. URL: https://doi.org/10.1007/s11263-009-0281-6. doi:10.1007/s11263-009-0281-6.

[39] Johnson, AE, Hebert, M. Surface matching for object recognition in complex three-dimensional scenes. Image and Vision Computing 1998;16(9):635–651. URL: https://www.sciencedirect.com/science/article/pii/S0262885698000742. doi:10.1016/S0262-8856(98)00074-2.

[40] Frome, A, Huber, D, Kolluri, R, Bülow, T, Malik, J. Recognizing Objects in Range Data Using Regional Point Descriptors. In: Pajdla, T, Matas, J, editors. Computer Vision - ECCV 2004. Lecture Notes in Computer Science; Berlin, Heidelberg: Springer. ISBN 978-3-540-24672-5; 2004, p. 224–237. doi:10.1007/978-3-540-24672-5_18.

[41] Guo, Y, Sohel, F, Bennamoun, M, Lu, M, Wan, J. Rotational Projection Statistics for 3D Local Surface Description and Object Recognition. International Journal of Computer Vision 2013;105(1):63–86. URL: https://doi.org/10.1007/s11263-013-0627-y. doi:10.1007/s11263-013-0627-y.

[42] Rusu, RB, Blodow, N, Beetz, M. Fast Point Feature Histograms (FPFH) for 3D registration. In: 2009 IEEE International Conference on Robotics and Automation. 2009, p. 3212–3217. doi:10.1109/ROBOT.2009.5152473; iSSN: 1050-4729.

[43] Berker Logoglu, K, Kalkan, S, Temizel, A. CoSPAIR: Colored Histograms of Spatial Concentric Surflet-Pairs for 3D object recognition. Robotics and Autonomous Systems 2016;75:558–570. URL: https://www.sciencedirect.com/science/article/pii/S0921889015002225. doi:10.1016/j.robot.2015.09.027.

[44] Fischer, K, Gärtner, B, Kutz, M. Fast Smallest-Enclosing-Ball Computation in High Dimensions. In: Di Battista, G, Zwick, U, editors. Algorithms - ESA 2003. Berlin, Heidelberg: Springer. ISBN 978-3-540-39658-1; 2003, p. 630–641. doi:10.1007/978-3-540-39658-1_57.

[45] Fischer, K. hbf/miniball. 2024. URL: https://github.com/hbf/miniball; original-date: 2013-02-20T11:00:46Z.

[46] Rouwe, J. Jolt Physics. 2024. URL: https://github.com/jrouwe/JoltPhysics; original-date: 2021-08-12T14:12:24Z.

[47] Mamou, K. Volumetric Hierarchical Approximate Convex Decomposition. In: Game Engine Gems, Volume 3; 1st edition ed. Boca Raton: A K Peters/CRC Press. ISBN 978-1-4987-5565-8; 2016, p. 141–158.

[48] Mammou, K, Ratcliff, JW. kmammou/v-hacd. 2024. URL: https://github.com/kmammou/v-hacd; original-date: 2015-03-15T03:20:46Z.

[49] Surazhsky, V, Gotsman, C. High quality compatible triangulations. Engineering with Computers 2004;20(2):147–156. URL: https://doi.org/10.1007/s00366-004-0282-6. doi:10.1007/s00366-004-0282-6.

[50] CGAL, Computational Geometry Algorithms Library. 1996. URL: https://www.cgal.org.

[51] van Blokland, BI, Theoharis, T. Partial 3D Object Retrieval using Local Binary QUICCI Descriptors and Dissimilarity Tree Indexing. arXiv:210703368 [cs] 2021;URL: http://arxiv.org/abs/2107.03368; arXiv: 2107.03368.

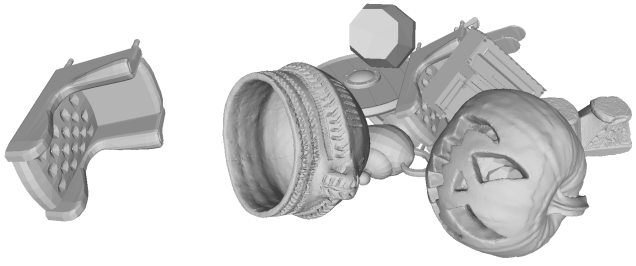
[52] van Blokland, BI. bartvbl/libShapeDescriptor. 2018. URL: https://github.com/bartvbl/libShapeDescriptor; original-date: 2018-10-05T07:51:40Z.

[53] Carmichael, O, Huber, D, Hebert, M. Large data sets and confusing scenes in 3-D surface matching and recognition. In: Second International Conference on 3-D Digital Imaging and Modeling (Cat. No.PR00062). 1999, p. 358–367. doi:10.1109/IM.1999.805366.

[54] Själander, M, Jahre, M, Tufte, G, Reissmann, N. EPIC: An Energy-Efficient, High-Performance GPGPU Computing Research Infrastructure. 2024. URL: http://arxiv.org/abs/1912.05848. doi:10.48550/arXiv.1912.05848; arXiv:1912.05848 [cs].

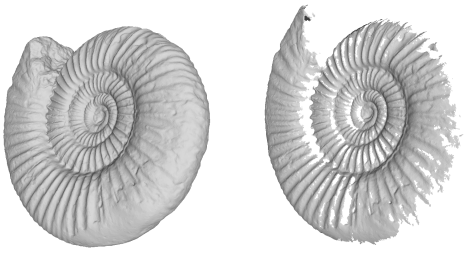
Supplementary Material

Examples of filtered objects



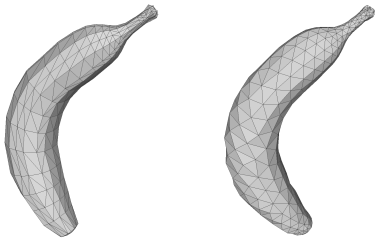
(a) Original (b) Filtered

Fig. 18: Example output of the clutter filter.



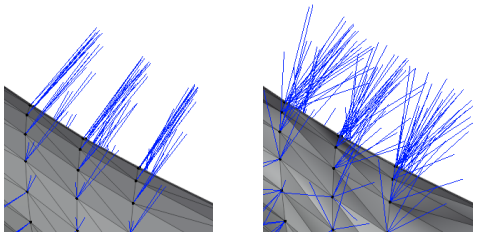
(a) Original (b) Filtered

Fig. 19: Example output of the occlusion filter.



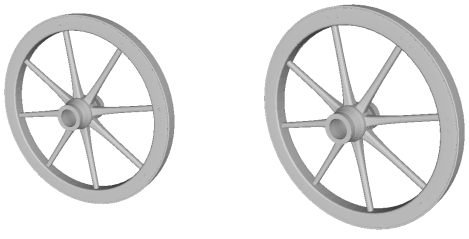
(a) Original (b) Filtered

Fig. 20: Example output of the alternate triangulation filter.



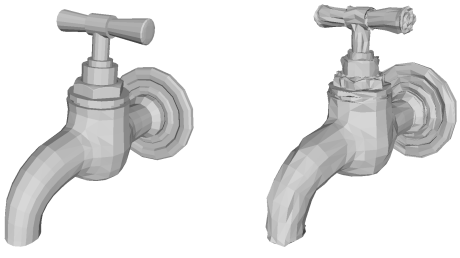
(a) Original (b) Filtered

Fig. 21: Example output of the normal vector deviation filter.



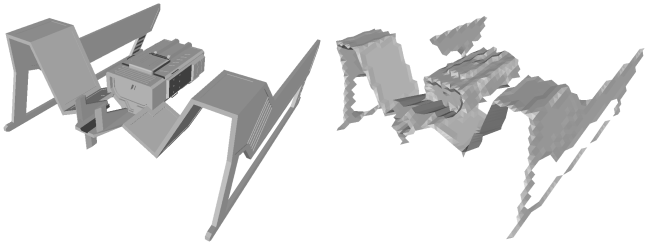
(a) Original (b) Filtered

Fig. 22: Example output of the support radius deviation filter.



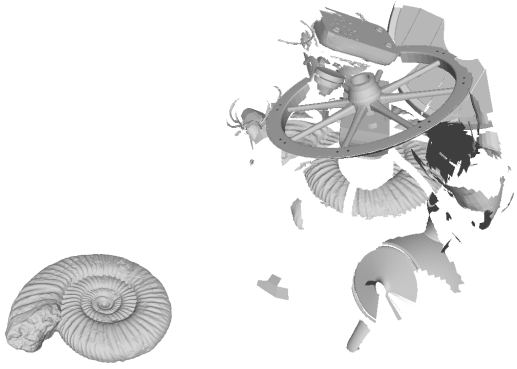
(a) Original (b) Filtered

Fig. 23: Example output of the Gaussian noise filter.



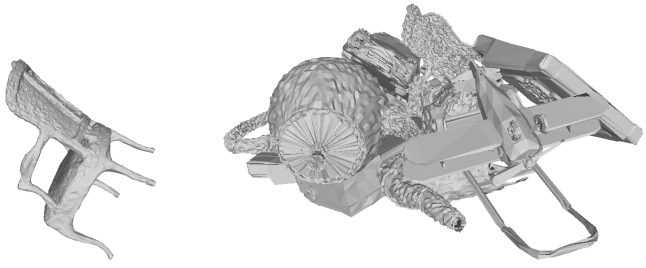
(a) Original (b) Filtered

Fig. 24: Example output of the alternate mesh resolution filter.



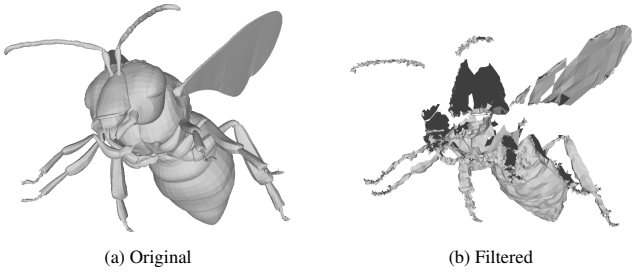
(a) Original (b) Filtered

Fig. 25: Example output of the clutter filter followed by the occlusion filter.



(a) Original (b) Filtered

Fig. 26: Example output of the clutter filter followed by the Gaussian noise filter.



(a) Original (b) Filtered

Fig. 27: Example output of the occlusion filter followed by the Gaussian noise filter.