Computers & Graphics 91 (2020) 118–128

Contents lists available at ScienceDirect

Computers & Graphics

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

Technical Section

Radial intersection count image: A clutter resistant 3D shape descriptor

Bart Iver van Blokland*, Theoharis Theoharis

Department of Computer Science, Norwegian University of Science and Technology (NTNU) Norway

ARTICLE INFO

Article history:
Received 1 November 2019
Revised 22 July 2020
Accepted 22 July 2020
Available online 28 July 2020

Keywords:
Computers and graphics
Formatting
Guidelines

ABSTRACT

A novel shape descriptor for cluttered scenes is presented, the Radial Intersection Count Image (RICI), and is shown to significantly outperform the classic Spin Image (SI) and 3D Shape Context (3DSC) in both uncluttered and, more significantly, cluttered scenes. It is also faster to compute and compare. The clutter resistance of the RICI is mainly due to the design of a novel distance function, capable of disregarding clutter to a great extent. As opposed to the SI and 3DSC, which both count point samples, the RICI uses intersection counts with the mesh surface, and is therefore noise-free. For efficient RICI construction, novel algorithms of general interest were developed. These include an efficient circle-triangle intersection algorithm and an algorithm for projecting a point into SI-like (α, β) coordinates. The 'clutterbox experiment' is also introduced as a better way of evaluating descriptors' response to clutter. The SI, 3DSC, and RICI are evaluated in this framework and the advantage of the RICI is clearly demonstrated.

1. Introduction

Local shape descriptors have seen extensive use in a wide variety of applications where determining shape correspondences are beneficial or even required. Such applications include registration [1–3], shape segmentation [4–6], and retrieval [7,8].

Many local 3D shape descriptor methods rely on the surfaces present in the volume around a point to compute the degree to which two points are similar. This also makes them susceptible to any unwanted geometry present in the neighbourhood, commonly referred to as *clutter*. For this reason, clutter has been named as a major factor degrading the performance of current descriptors [9].

The degree to which different descriptors are capable of resisting the negative effects of clutter varies. One classical method which has shown to be significantly resistant to clutter is the Spin Image [10] (SI). This descriptor is invariant under rigid transformations, and has been applied successfully for applications such as shape registration [11] and facial recognition [12].

In this paper, we present the Radial Intersection Count Image (RICI) combined with a novel distance function. The new descriptor shares the original concept of the Spin Image but is advantageous in terms of its generation speed and clutter resistance.

In order to show the effectiveness of the RICI, we propose a repeatable experiment aimed at quantifying the effects of clutter on the matching performance of 3D shape descriptors. The main advantage of this evaluation method is that it can be used with datasets of any size, and ensures scenes are cluttered with natural shapes.

In summary, the contributions of this paper are:

1. The novel RICI descriptor and an accompanying distance function, capable of resisting clutter.
2. Algorithms for efficient generation of RICI descriptors, also capable of accelerating SI construction.
3. The clutterbox experiment for quantifying the effects of clutter.
4. Evidence that the Support Angle filter proposed in the original SI paper does not necessarily improve matching performance.
5. Freely available GPU implementations for generating and comparing Spin Image, 3DSC, and RICI descriptors, as well as an implementation of the proposed clutterbox experiment.

2. Background and related work

Numerous local shape descriptors have been proposed to date [9]. The Spin Image has been the foundation for a number of methods, which attempt to improve its matching performance or other

This article has been certified as Replicable by the *Graphics Replicability Stamp Initiative:* http://www.replicabilitystamp.org

* Corresponding author.

E-mail addresses: bart.van.blokland@ntnu.no (B.I. van Blokland), theotheo@ntnu.no (T. Theoharis).

https://doi.org/10.1016/j.cag.2020.07.007

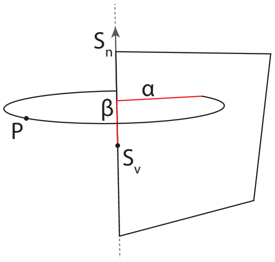


Fig. 1. A visualisation of the α and β coordinates corresponding to a given point P, relative to the Spin Vertex S_v and Spin Normal S_n. The Central Axis; the line described by the Spin Vertex and Spin Normal is also shown.

limitations. Clutter is a major challenge for object descriptors and few methods have addressed it.

2.1. Spin images

The Spin Image [10], originally presented by Johnson et al., is a classic descriptor generated from an oriented point cloud (vertices with position and normal).

An SI is constructed around an oriented point, the position of which is in this paper referred to as the Spin Vertex S_v. The corresponding normal is referred to as the Spin Normal S_n. The combined oriented point describes a line, which is called the Central Axis.

Computing the descriptor involves placing a square plane whose left side is on the Central Axis, with the Spin Vertex at its vertical halfway point. This plane is subsequently subdivided into ($N_{bins} \times N_{bins}$) equivalently sized bins, and rotated for one revolution around the Central Axis. As the plane rotates, the number of point samples intersecting each bin is counted. The descriptor itself is a histogram of the resulting value of each bin, which can be visualised as an image.

In practice, the locations where point samples will intersect with the rotating square can be computed directly as two-dimensional cylindrical coordinates. Here the α coordinate refers to the distance from the point sample to the closest point on the Central Axis, and the β coordinate refers to the distance from this closest point to the Spin Vertex. The projection of a given point P is shown in Fig. 1.

The physical width and height of the square plane is the Support Radius of the descriptor. By rotating the plane around the Central Axis, a cylindrical volume is created, which represents the Support Volume of the descriptor. Additionally, point sample contributions are divided over nearby bins using bilinear interpolation to reduce the effects of aliasing.

Johnson et al. also describe a prefiltering step called the Support Angle, where a sample oriented point is not included in the computation of the descriptor if the angle between its normal vector and the Spin Normal exceeds a set threshold.

The descriptor's core idea is that a pair of points with identical surfaces surrounding them, and assuming both have been uniformly sampled, will have proportional quantities of projected points in similar locations. Images can thus be compared using statistical correlation.

2.2. Methods related to the SI

One of the major issues with the Spin Image is its volatility. Uniform sampling of triangle meshes as well as scans from 3D capture devices are inherently noisy. Carmichael et al. proposed a method to address this by computing the exact area of the support region intersecting each pixel [13].

Other methods aim to address specific limitations of the spin image. Assfalg et al. proposed the spin image signature aimed at simplifying the ease of image retrieval from a large database [14]. Dinh et al. aimed at addressing the issue of selecting bin sizes by creating a spin image variant with variable sized histogram bins [15], although their solution involves the manual setting of parameters.

An alternate spin image variant, proposed by Guo et al. used three spin images per vertex rather than a single one for better matching performance [16]. Accelerating spin image generation using a GPU was first proposed by Davis et al. [17,18]. Alternate derivative methods include Spin Contours, proposed by Liang et al. [19] and colour spin images by Pasqualotto et al. [20].

2.3. The 3D shape context

The 3D Shape Context, proposed by Frome et al. [21], is a histogram descriptor constructed by accumulating points by their spherical coordinates and distance relative to an oriented reference point in a spherical support region. The support region is divided into J equally spaced spherical wedges, centred around the central axis described by the reference oriented point (similar to the SI). Each wedge is subsequently divided into K elevation divisions. The bin volumes are finally created by the intersection volume of each radial and elevation divisions with the volume bounded by two of L successive spheres with exponentially increasing radii.

The descriptor has a degree of freedom around the Central Axis, which the Authors solve by generating J different descriptors for each vertex, where each of the wedges has been offset by a multiple of the angle $\frac{2\pi}{J}$. However, due to its self-symmetry, this step is unnecessary for descriptors used for querying.

2.4. Other clutter-Resistant shape matching methods

Some methods which have been proposed to date, in addition to the Spin Image and 3DSC, have been shown to perform better in cluttered scenes than others [9,22].

Mian et al. presented a method which creates a three-dimensional grid of voxels based on two randomly selected vertices, referred to as a Tensor [22]. Their results outperform the Spin Image, and show resistance to clutter being present in the scene.

The THRIFT descriptor, proposed by Flint et al. [23], uses an approach similar to the Scale-Invariant Feature Transform (SIFT) by Lowe et al [24]. The method aims to find distinctive points which can be detected reliably under a wide range of conditions. This is accomplished by computing a three-dimensional density map of the input point cloud, and selects interest points by locating local maxima of the Hessian matrix.

Local surface patches, proposed by Chen et al. [25], is a two-dimensional histogram descriptor generated from points in an oriented point cloud. Each descriptor accumulates points in a spherical support volume, by their shape index and the cosine of the angles between their normal vectors. The authors only test their method on range images, and do not expose the descriptor to significant levels of clutter themselves. However, experiments performed in the review by Guo et al. [9] suggest that this method performs well in cluttered scenes.

Unfortunately, the above works on clutter resistant descriptors used very small datasets for testing their methods (1 to 56 objects). Therefore, the provided results may be statistically biased, since the proposed descriptors were not subjected to a sufficiently wide range of possible surface features. The datasets used were also not made public, making it difficult to compare their results. In addition, some used very similar objects (such as cars), presumably for ease of creation, which is not representative of all forms of clutter that can be encountered in a real scene.

2.5. *Learning approaches*

More recent shape matching methods have attempted to utilise Neural Networks. One of the major hurdles these methods need to overcome is the inherent irregularity present in 3D shape data, as opposed to more regular data such as images on which learning methods have been applied successfully.

To this end, many methods, such as the PPFNet proposed by Deng et al. [26], make use of existing descriptors or features in a pre-processing step to regularise the input to the neural network. PPFNet specifically uses point pair features, and was shown to outperform many current state-of-the-art handcrafted methods.

Another regularisation approach is the voxelisation of the input point cloud or mesh, which has amongst others been exploited in the 3DMatch method proposed by Zeng et al. [27], who successfully apply their proposed method on point cloud alignment and keypoint matching, outperforming both handcrafted and earlier learning methods.

While these learning methods show great promise, their applicability depends highly on the used dataset for training, and may require retraining for new environments. Moreover, current learning methods tend to be highly computationally expensive, which can limit their applicability to small datasets only [28].

3. Radial intersection count images (RICI)

The novel RICI descriptor is now detailed, which shares some conceptual similarities with the original Spin Image, and has preliminarily been proposed as a quasi Spin Image [29].

3.1. *RICI Generation*

A RICI descriptor is a 2D histogram of integers. It is constructed around an oriented point, and has a Central Axis around which a square plane is conceptually rotated, similar to the Spin Image. The square plane is divided into ($N_{bins} \times N_{bins}$) bins, producing a histogram which can be visualised as a grayscale image.

The primary difference between the RICI and the SI is what is counted in each histogram bin. In Spin Images, projected point samples are accumulated to create an estimate of the surface area intersecting each bin or pixel as the square plane is rotated for a full revolution. In contrast, RICI bins count the number of intersections of circles with the surfaces of the scene and are thus integers.

The conceptual construction method, i.e. the relationship between the aforementioned intersection circles and the produced descriptor is visualised in Fig. 3. Consider a set of circles that are centred at fixed distances from the Spin Vertex on the Central Axis and have a fixed number of radii. Each bin in the RICI image stores the number of intersections of the corresponding circle with the surfaces of the scene. RICI rows thus represent circles on the same plane, and RICI columns circles with equivalent radii.

The remainder of this section presents a method for efficiently computing RICI descriptors. The general idea is to iterate over each triangle in the scene, and determine the set of circles in cylindrical coordinates (see Fig. 1) which will intersect with it. This implies a complexity of O(T), where T is the number of triangles in the scene, as in the worst case, the number of circles is fixed and equal to the resolution of a RICI image. The bins corresponding to these circles are incremented. Note that cylindrical projections will not preserve the linearity of a triangle's edges (as shown in Fig. 2), thus not allowing the use of common rasterisation methods. Instead we exploit a circle-triangle intersection algorithm in order to determine the correct projections.

To summarise, a RICI image is generated by iterating over each triangle in the scene, and in turn each triangle is processed in 3 steps:

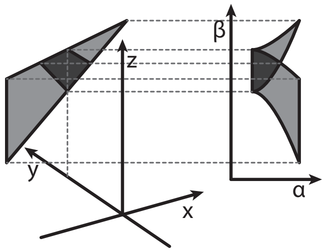


Fig. 2. A triangle depicted alongside its projection in cylindrical coordinate space. The area in which circles centred and directed along the z-axis intersect the triangle twice is coloured in dark grey. Sizes may not be to scale.

1. Project the triangle vertices into cylindrical coordinate space, as described in Section 3.1.1.
2. Using the circle-triangle intersection method outlined in Section 3.1.2, compute the range of α coordinates which will intersect with the triangle for each β coordinate in the triangle's β-extent.
3. Increment the histogram bins that correspond to these intersections.

3.1.1. *Projecting vertices into cylindrical coordinate space*

An efficient method for projecting points from Euclidean coordinates into cylindrical coordinates is presented. Apart from the RICI, this method can also be applied directly in the construction of SI descriptors.

The algorithm projects a point $P = (P_x, P_y, P_z)$ by computing two transformations. First, a translation that moves the Spin Vertex $S_v = (S_{vx}, S_{vy}, S_{vz})$ to the origin (Eq. 3), and second, a rotation which aligns the Spin Normal $S_n = (S_{nx}, S_{ny}, S_{nz})$ with the z-axis. The projected point's α and β coordinates can be computed trivially afterwards.

For the z-axis alignment transformation, a common technique for aligning two vectors consists of a vector product followed by a rotation (shown in Fig. 4). While the vector product itself is inexpensive (due to one of the vectors being the z-axis) the subsequent alignment rotation requires a relatively expensive multiplication with a 3×3 matrix.

Our alignment method instead uses two rotations, exploiting the observation that only distance must be preserved for the α coordinate. We align the spin normal with the xz-plane using a rotation around the z-axis (see Fig. 5(a) and Eq. 4). We then align the transformed normal with the z-axis by a rotation around the y-axis (Fig. 5(b) and Eq. 5).

$$[N_{ax}, N_{ay}] = Normalize[S_{nx}, S_{ny}]$$
$$[N_{bx}, N_{bz}] = Normalize[S_{nx}, S_{nz}] \tag{1}$$

$$P'_x = P_x - S_{vx}$$
$$P'_y = P_y - S_{vy} \tag{2}$$
$$P'_z = P_z - S_{vz}$$

$$P''_x = N_{ax} \cdot P'_x + N_{ay} \cdot P'_y$$
$$P''_y = -N_{ay} \cdot P'_x + N_{ax} \cdot P'_y \tag{3}$$

$$T_x = N_{bz} \cdot P''_x - N_{bx} \cdot P'_z$$
$$T_y = P''_y \tag{4}$$
$$T_z = N_{bx} \cdot P''_x + N_{bz} \cdot P'_z$$

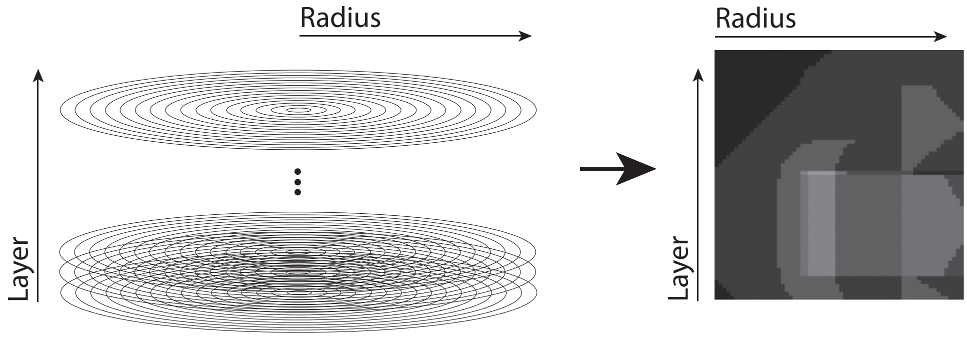


Fig. 3. A visualisation of the construction of a RICI image.

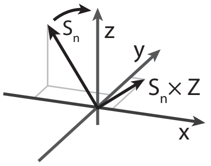
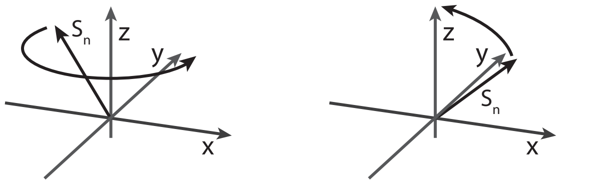


Fig. 4. Direct approach for vector alignment. First, compute the vector product $S_n \times Z$ between the spin normal S_n and z-axis. Second, rotate S_n around $S_n \times Z$ to align it with the z-axis.



(a) Rotation 1: Align the spin normal with the XZ-plane by a rotation around the Z-axis (Equation 3)

(b) Rotation 2: Align the spin normal with the Z-axis by a rotation around the Y-axis (Equation 4)

Fig. 5. Visual representation of the rotations that form our alignment method.

$$\alpha_i = |(T_x, T_y)|$$
$$\beta_i = T_z \quad (5)$$

The coefficients of the rotation transformations N_a and N_b can be calculated inexpensively from components of the spin normal S_n, as shown in Eq. 1. When both coefficients of either N_a or N_b are zero, that rotation step is unnecessary and an identity rotation is used instead. The key here is that, considering a two-dimensional coordinate system *xy*, the coordinates of a normalised vector represent the sine and cosine values of a rotation which aligns that vector with the x-axis. These normalised coordinates can therefore be used directly for this purpose.

It should be noted that since the rotation coefficients only depend on the spin normal, they are constant for the entire spin image. Therefore they only need to be computed once per image, essentially taking this computation out of the inner loop. This is the primary reason for the method's efficiency compared to previous work.

3.1.2. Circle-Triangle intersection

A circle-triangle intersection test can result in four outcomes; no intersection, one intersection, two intersections, or infinite intersections. However, due to floating point rounding errors, handling the latter, while possible, is not feasible in practice and is thus not addressed by the proposed algorithm.

Our algorithm starts off with the triangle vertices in cylindrical coordinate space. For a given β coordinate, it determines the range of α coordinates which result in a single or double intersection. This information is subsequently used to "rasterise" a row of pixels for the triangle in the RICI descriptor.

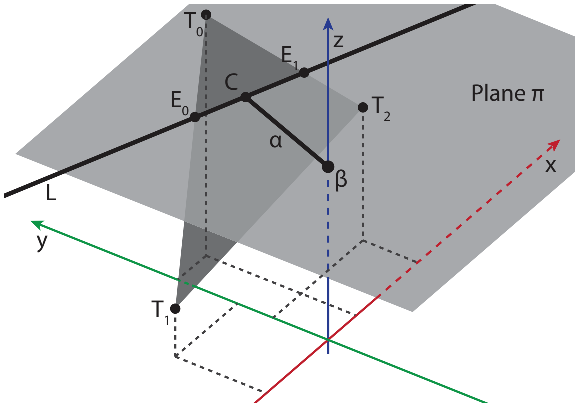


Fig. 6. A triangle defined by the vertices T_0, T_1, and T_2 intersecting with the horizontal plane through an arbitrary coordinate β on the z axis.

The method operates in three distinct stages. First, the triangle is intersected with the plane π of the circle, which is parallel to the *xy* plane, as shown in Fig. 6. Next, the triangle vertices are rotated around the z-axis in order to further simplify subsequent computations. Finally, the ranges of circle radii in which respectively single and double intersections occur, are calculated.

Prior to detailing these stages individually, we will outline the geometric background used in the intersection test calculations.

Fig. 6 shows a given β coordinate. The triangle being tested is defined by its transformed vertices T_0, T_1, and T_2, using the previously described alignment transformation. Here all points with equal β coordinates lie on the plane π.

Where the triangle intersects the plane, it forms an intersection line segment E_0E_1, which defines a line *L*. The range of α coordinates either intersecting the triangle once or twice can be calculated by determining which radii intersect with this line segment. This reduces the determination of intersection distances to a two-dimensional problem.

For single intersections, the lower and upper bounds of radii is $[min(|E_0|, |E_1|), \max(|E_0|, |E_1|)]$. Note that the 2D coordinates of E_0 and E_1 are equivalent to the vectors $\beta\vec{E}_0$ and $\beta\vec{E}_1$, respectively.

A double intersection occurs when the closest point to β on line *L* is also on the line segment E_0 E_1. When double intersections exist, the range of radii in which they occur is $[|C|, \min(|E_0|, |E_1|)]$.

Given the aforementioned background, the next step of our method is aligning the vector $\vec{\beta C}$ with the y-axis, as illustrated in Fig. 7. The objective of this step is to simplify the remaining calculations for the intersection test. Alignment is done by normalising the vector between E_0 and E_1, and subsequently rotating the triangle vertices around the z-axis; the coordinates of the normalised

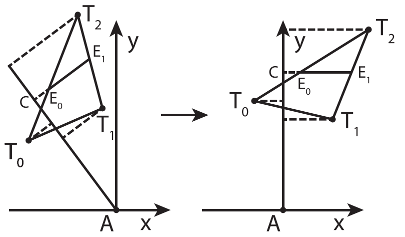


Fig. 7. Aligning an $E_0\vec{E}_1$ vector with the x-axis. Any value of C can be chosen for which an $E_0\vec{E}_1$ vector exists for this purpose. A sample $E_0\vec{E}_1$ vector has been indicated in the Figure. Point A represents the point on the Central Axis marked by β in Fig. 6.

vector can be used directly as sine and cosine coefficients for the rotation.

At this stage, determining the existence of a double intersection is inexpensive, and can be achieved by comparing signs of the x components of the aligned E_0 and E_1 coordinates. Different signs indicate that a double intersection exists. If so, the length of $\overrightarrow{\beta C}$ (the rotated y-coordinate of C) represents the lower bound of radii which correspond to double intersections.

The intersection test itself can be done by comparing a given radius against the computed ranges, which yields an intersection count corresponding to that radius.

Summarising, computing the range of values of α that will result in a single or double intersection for a given value of β involves the following steps:

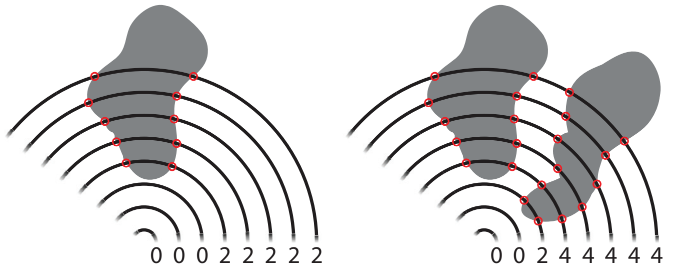
1. Determine the intersection points E_0 and E_1 for any value of value of β where L is defined, as shown in Fig. 6.
2. Rotate E_0 and E_1 around the z-axis such that the vector $E_0\vec{E}_1$ is aligned with the x-axis (as shown in Fig. 7).
3. Determine the distance of E_0 and E_1 from the z-axis.
4. The range of circle radii in which single intersections occur is $[min(|E_0|, |E_1|), \max(|E_0|, |E_1|)]$.
5. Determine the existence of a double intersection by comparing the signs of the x-coordinates of E_0 and E_1. If they are different then a double intersection exists.
6. If a double intersection exists, the range of α coordinates (circle radii) corresponding to the double intersection is the y-coordinate of either E_0 or E_1 and the shortest distance between the z-axis and E_0 or E_1.

3.2. A Clutter-Resistant RICI Distance function

Spin Images, by their nature of being generated from oriented point clouds, are inherently noisy. They have as such relied on statistical correlation to compute similarity. The idea here is that two matching bins tend to have proportionally similar accumulated sample counts. Unfortunately, this method is susceptible to the effects of clutter. Additional geometry present in the support volume causes portions of the image to receive additional projected point samples, which consequently negatively affects the computed correlation value.

When it comes to comparing RICIs, one important downside of the Pearson Correlation Coefficient is that it is not defined for sequences of constant values. While this scenario is unlikely to occur for Spin Images, there exist situations in which RICIs consist solely of pixels with equivalent intersection counts. For these situations, the Pearson correlation coefficient is undefined, and therefore an insufficient solution for comparing RICIs. Handling these edge cases separately is possible, but results in a solution that requires balancing awarded scores against normal situations.

Meanwhile, the RICI does not have the aforementioned issue of noise, and is as such not bound solely to using statistical methods



(a) Intersection counts without clutter

(b) Intersection counts with clutter

Fig. 8. Demonstration of changes in intersection counts generally being unaffected by clutter. A portion of a single layer of intersection circles is shown. Intersections with the shape surface have been marked.

for measuring similarity. For these reasons we propose a new distance function, which is by design able to resist some of the negative effects of clutter, primarily by exploiting features of the RICI.

First, the distance function does not consider the values of pixels in the RICI. Instead, *changes* in pixel values (i.e. intersection counts which show up as edges in the RICI) are compared. As RICIs are free of noise, it is possible to interpret pixel values directly. The main advantage of this approach is that changes in intersection counts are largely unaffected by clutter. The reason for this can be seen in Fig. 8.

In Fig. 8(a), a cross section is shown of an arbitrary 3D shape. On the same plane, circles are drawn with increasing radii, similar to how RICI images are computed. The numbers below each circle indicate the number of intersections they encounter, which corresponds to the value of their respective pixels in the RICI image.

Similarly, Fig. 8(b) shows the same situation in which a clutter object has been added. From the intersection counts can be seen that even though the absolute intersection counts have now changed, the change in intersection counts from the third to the fourth circle, caused by the original object, is still present.

Second, when searching, our distance function treats the *needle* (query) and the *haystack* image asymmetrically, in contrast to the Pearson correlation coefficient. One can use the needle image to deduce what features to look for in a given haystack image.

This asymmetry consists of only computing a sum of squared differences distance on pixels where there are *changes* in the needle RICI image.

Returning to Fig. 8, we'll assume that Fig. 8(a) shows a cross section of the needle object that we are attempting to locate in the cluttered haystack scene shown in Fig. 8(b). In our needle image, only the increased intersection counts from the third to the fourth circle are relevant. Including other pixels is not relevant, as there are no changes in the needle image's intersection counts. We can therefore ignore these pixels in our distance computation. This also means any clutter present in the haystack image is ignored by this method.

The proposed Clutter Resistant Distance function $CRD(needleRICI, haystackRICI)$ is shown in Eq. 7, and the corresponding pseudocode is given in Listing 1. Note here that the distance function is positive, but not symmetric. It has a complexity of O(1), because comparing a descriptor pair requires a fixed number of operations.

$$D(rici, r, c) = rici(r, c) - rici(r, c-1) \tag{6}$$

$$CRD(n, h) = \sum_{r=0}^{N_{bins}} \sum_{c=1}^{N_{bins}} \begin{cases} (D(n, r, c) - D(h, r, c))^2, & \text{if } D(n, r, c) \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

```
def clutterResistantDistance(needle, haystack):
  score = 0
  for row r in [0..N_bins]:
    # Skip first column
    for column c in [1..N_bins]:
      needleDelta =
            needle[r][c] - needle[r][c-1]
      haystackDelta =
            haystack[r][c] - haystack[r][c-1]
      if needleDelta != 0:
        score +=
            (needleDelta - haystackDelta) *
            (needleDelta - haystackDelta)
  return score
```

Listing 1. Pseudocode for our proposed method for computing the distance between two RICI images.

4. Evaluation

The proposed method has been evaluated in terms of its clutter resistance, generation speed, and matching performance. Where applicable, we compare our method against the two most referenced among those listed in survey [9] as being clutter resistant. These are the Spin Image[1] and the 3D Shape Context. It is worth noting that the survey also observes that popular descriptors such as the Fast Point Feature Histogram [32], Unique Signatures of Histograms [30], and Rotational Projection Statistics [31], do not exhibit optimal performance under cluttered conditions. We have therefore implemented the above two most referenced clutter resistant methods on the GPU, to allow a direct comparison on the same dataset.

The novel Clutterbox Experiment is proposed in order to evaluate the effect of clutter on the descriptors' matching performance.

4.1. The clutterbox experiment

In previous work, clutter has typically been defined as the proportion of area within the support volume that does not belong to the object being recognised. Greater proportions of clutter generally imply worse descriptor performance. The expression used in previous work, initially proposed by Johnson et al. [10] is shown in Eq. 8. Here A_{all} is the surface area of all objects within the support volume and A_{object} is the surface area of the object of interest.

$$clutter = \frac{A_{all} - A_{object}}{A_{all}} \tag{8}$$

The objective of the proposed evaluation method, which we call the "clutterbox experiment", is to measure the relationship between increasing levels of clutter and the resulting performance of the descriptor being tested.

In previous clutter experiments, clutter has generally been evaluated by measuring descriptor performance against levels of clutter present at points in a scene without controlling the points' identities. However, this measures the effects of two parameters combined; the descriptor's ability to recognise the desired shape, and the level of clutter present around it. Ideally an evaluation of the effects of clutter should control the former of these parameters, while varying the latter. This is the primary objective that the clutterbox experiment addresses.

Varying clutter levels in the neighbourhood of an object can be done trivially by adding triangles, points, spheres, or cubes in random locations and sizes around an object. However, this kind of clutter is not representative of the clutter that can be expected in a realistic 3D scene. The clutterbox experiment therefore inserts complete objects rather than random noise. This results in a more natural distribution of clutter in the scene, and therefore more directly measures the effect of clutter that can be expected of a given descriptor when applied in a practical context.

The clutterbox experiment is executed a large number of times by varying objects and their transformations, in order to provide robust results, independent of object type.

The steps of the experiment are outlined below:

1. Define the clutterbox as a cube of side s.
2. Select n objects at random from a large object collection.
3. Scale and translate each object such that it fits exactly inside a unit sphere.
4. Pick one of the n objects at random. This is the reference object.
5. Compute the reference descriptor set $\{RD\}$, by computing one descriptor for each unique vertex of the reference object.
6. For each of the n objects in random order, but starting with the reference object:
   - 6.1 Place the object within the clutterbox, at a randomly chosen orientation and position, with the constraint that the bounding sphere fits entirely within the clutterbox.
   - 6.2 Compute the set of cluttered descriptors $\{CD\}$, by computing one descriptor for each unique vertex of the combined mesh in the clutterbox.
   - 6.3 For each $d \in \{RD\}$, create a list of ranked distances to all $c \in \{CD\}$. Keep the rank where the corresponding cluttered descriptor was found in the ranked list ($0 \leq rank \leq |\{CD\}| - 1$). Note that lower ranks are better.
   - 6.4 Create a histogram where bin i holds the number of times the correct vertex is found in the search results at rank i.

Thus the output of the clutterbox experiment is a list of histograms, one for each level of clutter. A visualisation of a sequence of scenes with increasing clutter generated by the above experiment is shown in Fig. 9.

4.2. Clutter resistance evaluation

We used the clutterbox experiment to quantify the effects of clutter on the SI and 3DSC versus the proposed RICI descriptor. For our object collection, we selected the combined SHREC2017 dataset [33], which consists of 51,162 triangle meshes.

In the case of the SI and 3DSC, the combined triangle mesh of the reference and clutter objects was sampled into a point cloud before generating their descriptors; RICI descriptors are generated from the triangle mesh directly. For optimal performance, SI and 3DSC require a high number of samples to ensure a low level of noise in the produced descriptors. However, one cannot increase the sample count indefinitely as that results in a lower generation rate. Based on our experimental evidence on the given dataset, we feel that 10 samples per triangle is a reasonable point on this trade-off.

While Johnson et al. define the bin size (thus the support radius) of the SI to be equal to the mesh resolution, we do not believe their reasoning holds any longer for present day 3D objects. Similar objects can have significant variance in their resolution. As such, making the support radius dependent on the mesh resolution is not a guarantee for better matching performance. We therefore use a constant support radius for all tested methods, set to 0.3 units, relative to the bounding unit sphere, for all scenes in the experiment for ease of reproducibility. For the 3DSC, we set the minimum support radius to $r_{\min} = 0.048$ units, which is proportionally the same as the one originally used by Frome et al. [21].

[1] [30] and [31] also support the SI as a clutter resistant descriptor.

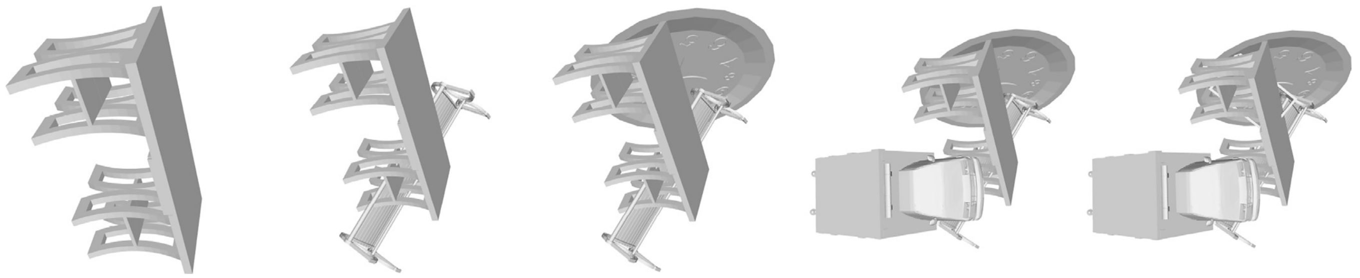


Fig. 9. Visual representation of the increasing number of clutter objects added into the clutterbox. The leftmost image only contains the reference object.

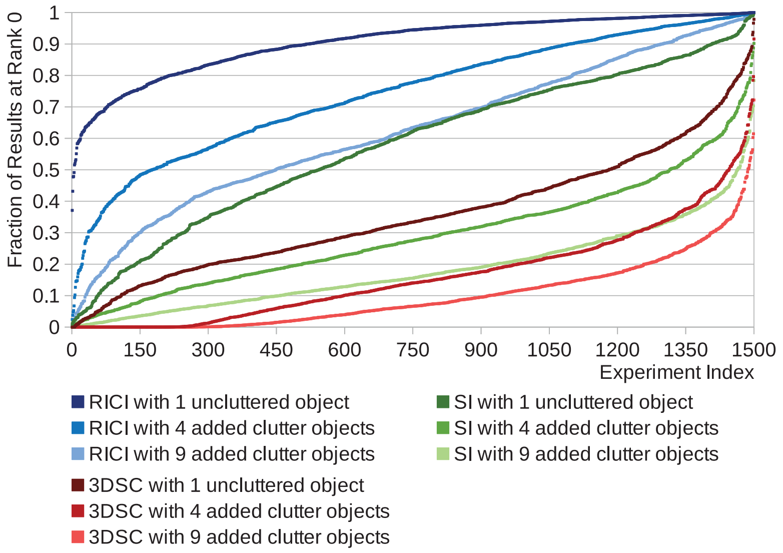


Fig. 10. Percentage of search results for all tested methods that ended up at rank 0 for each of the 1500 performed experiments.

We executed the experiment 1500 times, iteratively cluttering a scene with $n = 1$ (the reference object only), $n = 5$, and $n = 10$ objects, into a clutterbox of size $s = 3$. The size of the RICI and SI descriptors N_{bins} was set to 64x64 bins, while the 3DSC descriptor's dimensions were left the same as those used in previous work ($J = 15$, $K = 11$, $L = 12$ [9] [21]). A more detailed discussion on size settings can be found in Section 5.2. In order to visualise the histograms generated by the clutterbox experiment, we opted to compute the fraction of the bin representing rank 0 in the histogram against the sum of all bins (all search results). For clarity, each sequence of such fractions has been sorted individually to produce monotonically increasing curves. The results are shown in Fig. 10.

The support angle parameter used to generate the SI results in Fig. 10 requires further elaboration. In their original SI paper, Johnson et al. claim this filter reduces the effects of self-occlusion and clutter. However, our testing which compared using a support angle filter to not filtering any input points (Fig. 11) could not confirm this. All SI results in this paper therefore do not apply any support angle filter, as this favours the SI.

While Fig. 10 shows that our RICI descriptor clearly outperforms both the SI and 3DSC in scenes that contain clutter (see Eq. 8), it is also relevant to gain insight in the relationship between descriptor performance and the specific clutter level present in the support region. Fig. 12 shows a heatmap plot of the fractional area of clutter present in the support volume around each Spin Vertex, versus the rank of the corresponding descriptor in the haystack. It can be observed that the RICI trends towards lower ranks than the SI and 3DSC, even at high levels of clutter. Furthermore, while the 3DSC generally does not outperform the SI, it appears more clutter resistant than the SI at extreme clutter levels ($> 90\%$).

The heatmaps have been computed over 73.5 million search results extracted from scenes with 4 added clutter objects, based on the results of the Clutterbox experiment.

It is not expected that a RICI image would be very dependent on mesh resolution (which may be related to scanning) as intersection counts should in most cases not be very sensitive to that.

The experiment was implemented using C++, with the descriptor generation and search kernels written in CUDA 10.0. The code was written in such a way that given a dataset of objects, a single random seed determines all randomly chosen parameters, making all results reproducible. The experiment was executed on a combination of Nvidia Tesla cards (P100 16GB, V100 16GB, and V100 SXM3 32GB). All time-based results were exclusively gathered on the latter. One relevant implementation detail is that in cases where multiple search results have the same distance (which may occur due to reasons such as object self-similarity), we use the highest (best) rank of the matched haystack image for the sake of consistency.

4.3. Generation performance

Fig. 13 shows the difference in the rate at which the RICI, SI, and 3DSC descriptors are generated. As can be seen, the RICI is approximately one order of magnitude faster than the 3DSC, and two orders faster than the SI for the given settings.

4.3.1. Performance of point projection algorithm

The largest portion of the computational effort involved in the RICI and SI generation algorithms require projecting points into cylindrical coordinate space. We have proposed an efficient algorithm for this, as outlined in Section 3.1.1.

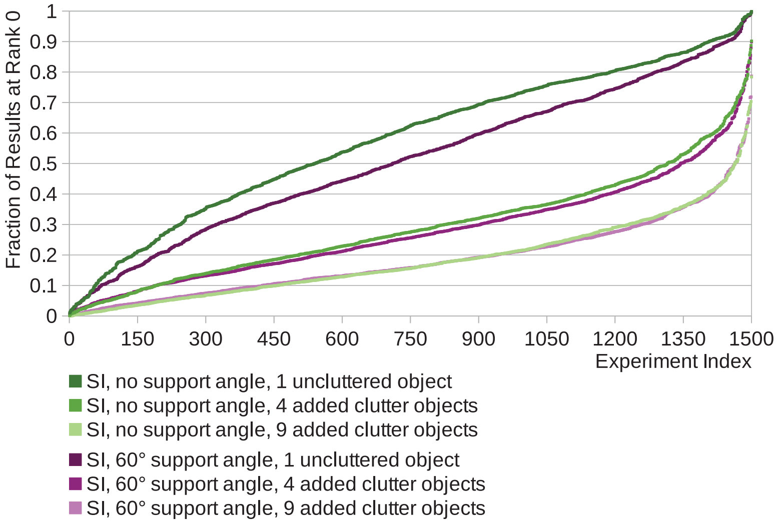


Fig. 11. Percentage of SI search results that ended up at rank 0 for each of the 1500 performed experiments for two different support angles.

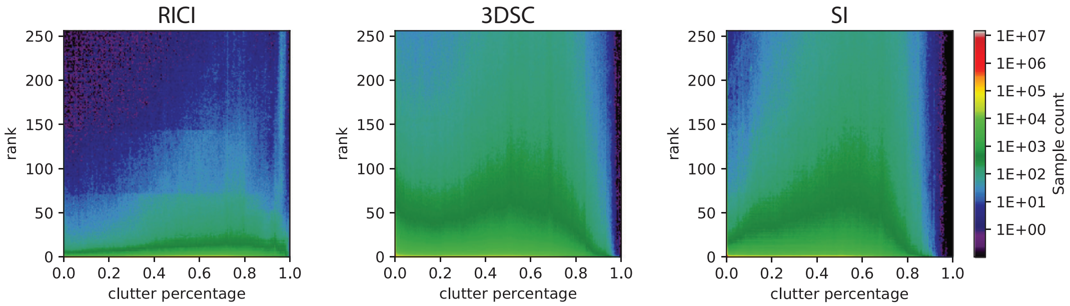


Fig. 12. Visualisation of the clutter resistance of RICI, SI, and 3DSC. Colours are mapped using a logarithmic function (colours toward the red end of the spectrum lower in the images is better). A pixel's colour represents the number of search results, i.e. descriptors, that ended up in the specific rank in relation to the amount of clutter within their support volume. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

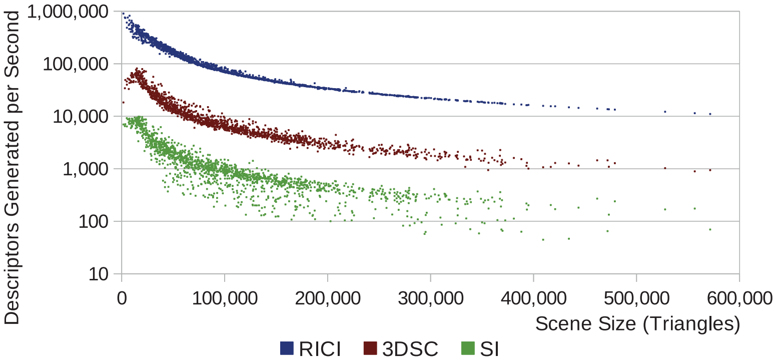


Fig. 13. Relationship between the number of triangles present in the scene, and the rate at which our implementations generate RICI, SI, and 3DSC descriptors.

A similar algorithm is included in Point Cloud Library [32], as part of the Spin Image generation implementation. To the best of our knowledge, this was up to now the most efficient implementation available. We therefore compare our projection algorithm against this previous work.

We evaluate both algorithms using a microbenchmark which projects a sequence of $1 \cdot 10^9$ randomly generated points. To ensure a fair comparison, all code unrelated to point projection has been removed from the Point Cloud Library SI generation implementation. The results are shown in Table 1.

It's worth noting that points are projected into cylindrical coordinates relative to the same oriented point. Our method can therefore precompute the values of N_{ax}, N_{ay}, N_{bx}, and N_{bz}, as outlined in Section 3.1.1. Both methods were tested on an Intel Core i7-8750H CPU.

Table 1
Point projection algorithm average execution times for projecting $1 \cdot 10^9$ points.

PCL (s)	Proposed method (s)
7.559	3.084

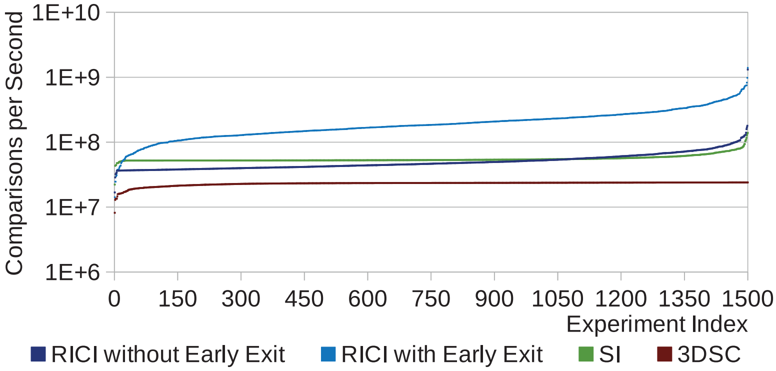


Fig. 14. Image matching rates in a scene with 5 objects. For clarity, each sequence has been sorted individually to produce a monotonically increasing curve.

4.4. Matching rate

The rates of evaluating the distance functions for each method are shown in Fig. 14. As can be seen, the RICI distance function's execution times are similar to the SI's Pearson correlation coefficient, while 3DSC is significantly slower.

For all methods, the bandwidth of the GPU memory bus is the main factor limiting the comparison rate. As our proposed distance function relies on computing the difference between neighbouring pixels, this would in a naive implementation, have required double the bandwidth. Instead, we use specialised “shuffle instructions” to read the value of neighbouring pixels without having to resort to another memory transaction, thereby halving the needed memory bandwidth. The result is a kernel whose memory bandwidth requirements, and consequently execution time, is similar to the Pearson Correlation Coefficient used to compare Spin Images.

We further optimised our implementation by using an early exit condition. Since the distance score can only go up for every subsequent pixel being processed, if the only objective is determining whether the distance between two images is smaller than some given threshold distance (as is the case in many retrieval applications), it is possible to cease execution when a predetermined distance threshold is exceeded. In our clutterbox experiment, this threshold can be trivially precomputed. Utilising this early exit condition resulted on average in a 4.2 times speedup over the SI distance function.

5. Observations and discussion

There are several topics and observations that may be relevant for the interpretation of the presented results.

5.1. Analysis of experimental results

While analysing the results presented in Section 4, we made several observations that are relevant to their interpretation. Fig. 15 contains a visualisation of a subset of these.

Fig. 15(a) shows the result set where RICI experienced the smallest decrease in matching performance between 0 and 9 added clutter objects in the scene. It is also possible to observe the clutter resistant properties of RICI. The seat part of the desk chair is significantly cluttered, while the wheels experience relatively small amounts of clutter (and remain visible). All three methods are capable of reasonably recognising these exposed wheels, however, the SI and 3DSC descriptors in large part fail to recognise the cluttered seat part.

Fig. 15(b) shows the result set where RICI experienced the largest drop in performance between the scenes with 0 and 9 added clutter objects. The primary cause of this drop is due to the cuboid-like shape and low level of details on the police van, which causes a low number of changes in intersection counts. In turn, the produced RICI images become relatively susceptible to clutter.

Fig. 15(c) shows the experiment where RICI performed worst on the uncluttered reference object. The particular object, a bookshelf, has high levels of self-similarity; a property which is also, to varying degrees, present in other objects in the CAD-oriented SHREC2017 dataset. Thus any local descriptor would rank vertices belonging to self-similar regions equally and whether they end up at Rank 0 is a matter of luck. One would expect to find them within the top *s* ranks, where *s* is the number of self-similar vertices. On the other hand, this is a useful tool for detecting self-similar regions.

To investigate this further we visualised the results of an experiment where the reference object had countable symmetric features, as shown in Fig. 16. As opposed to Fig. 15, we highlighted in red those vertices that were detected in the top *s* ranks instead of only rank 0. For instance, vertices in the table's legs are expected to constitute 12 self-similar partitions (6 legs with a symmetric front and backside each), which are all detected in the top 12 results, as shown in Fig. 16(d). Also all vertices in the base of the tabletop are correctly detected within the top 4 results (4-way symmetry).

In contrast to Fig. 15(c), (d) shows the experiment in which RICI had the highest recognition rate in the uncluttered scene. Little matching performance is lost after adding significant amounts of clutter.

In Fig. 15(e) the experiment whose drop in matching performance was closest to the total average of all performed 1500 experiments is shown. Worth noting here is the relatively low drop in recognition performance between the uncluttered scene, and the scene with 9 added clutter objects.

Finally, in Fig. 15(f) a rare phenomenon is shown where matching performance slightly improves between 4 and 9 added clutter objects.

5.2. Performance of 3DSC

As can be seen in Fig. 10, in contrast to the results obtained in previous work [9,21], the SI generally outperforms the 3DSC descriptor. The primary cause of this is that in previous work, the SI resolution was set to the 15x15 bins used originally by Johnson et al. [10]. In contrast, we used a resolution of 64x64 bins for parity with the RICI descriptor, which we also consider to be a resolution more suitable to the capabilities of modern processors. This significant increase in resolution meant the SI descriptor in our testing performs better than 3DSC with our chosen settings.

The decision to use the same bin dimensions for 3DSC as in previous work was primarily motivated by a tradeoff between

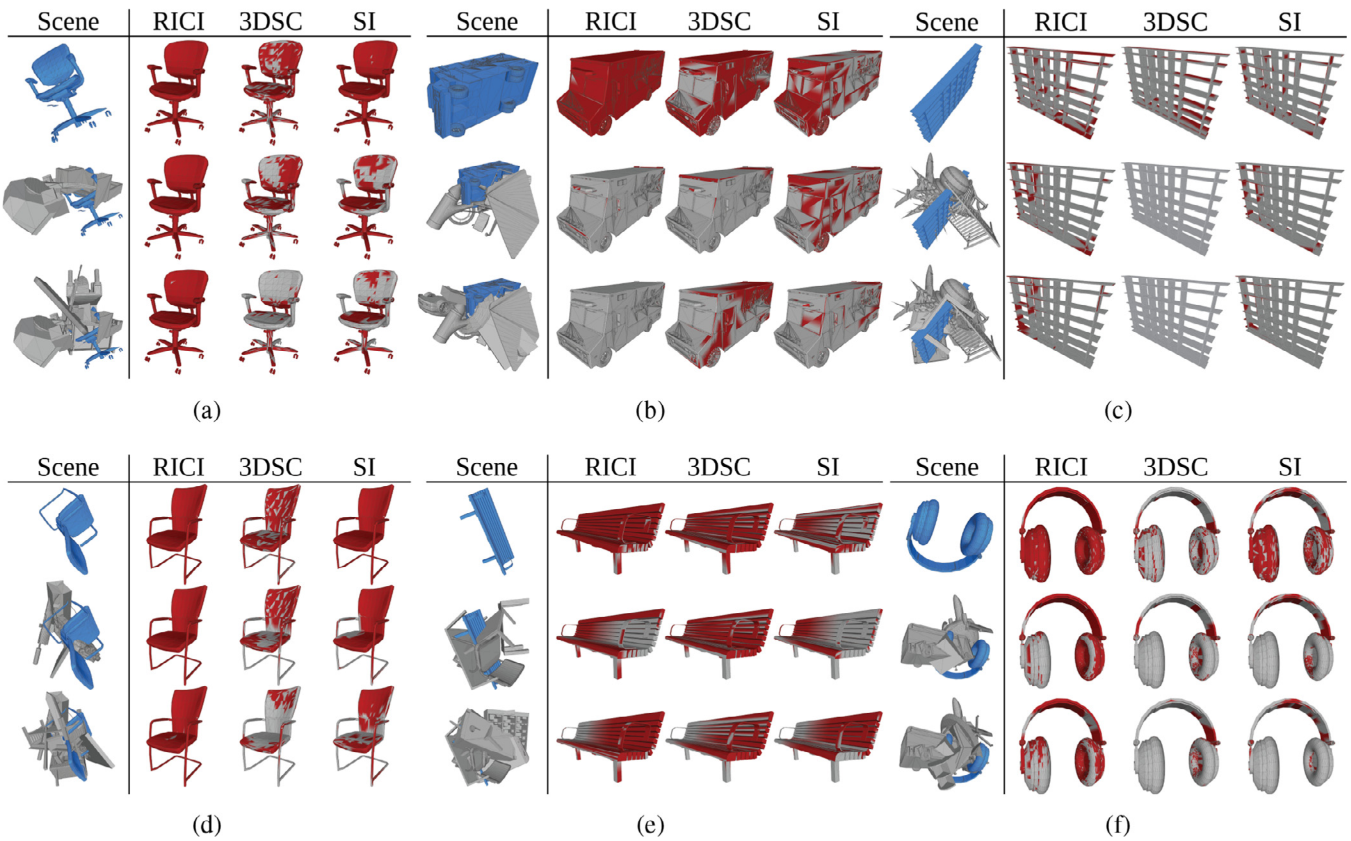
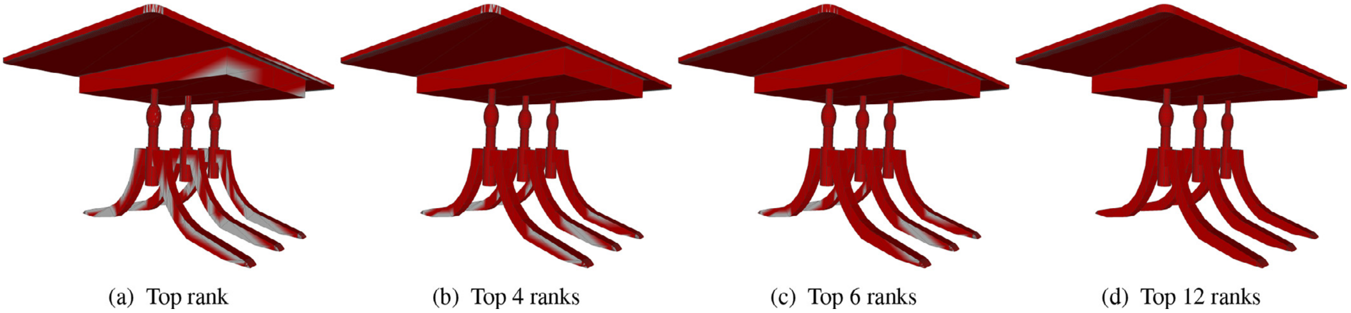


Fig. 15. Visualised results from 6 selected experiments. For each of the 6 subfigures, the Clutterbox scene (with 1, 5, and 10 objects) is shown on the left hand side, with the reference object highlighted in blue. Vertices correctly ranked at index 0 are highlighted in red, other vertices are coloured grey. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Top rank (b) Top 4 ranks (c) Top 6 ranks (d) Top 12 ranks

Fig. 16. Symmetric object whose vertices were present in the top s ranks of the search results, with varying values of s.

comparison performance and GPU hardware limitations. Our implementation makes use of shared memory when comparing 3DSC descriptors, due to the needle and haystack descriptor both being accessed once for each radial division. Current GPU shared memory pools allow fitting of approximately 2 image pairs sized at default settings simultaneously, which implies the number of bins can either be left intact, or doubled, or performance can be expected to be suboptimal. While it would be possible to double the number of bins in the 3DSC descriptor (which would make its memory requirements equal to the SI and RICI) leading to an increase in matching performance, the matching rate would decrease below acceptable levels because of the distance algorithm used. We therefore consider the used settings to be the best balance between quality and execution time for 3DSC.

6. Conclusion

In this paper, a clutter resistant shape descriptor, RICI, is presented and evaluated using a novel evaluation framework for such descriptors, called the clutterbox experiment. Novel algorithms for cylindrical coordinate projection, circle-triangle intersection, and the rasterization of triangles in cylindrical coordinates were presented. The largest quantitative evaluation of the SI, 3DSC, and RICI methods to date is also made, along with a useful observation for the SI support angle.

The main advantages of RICI are its noise-free nature and generation speed, while the related distance function makes it clutter resistant. We anticipate that the proposed clutterbox experiment, which is being made public, will aid future benchmarking of shape descriptors for cluttered scenes.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Bart Iver van Blokland: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Theoharis Theoharis:** Methodology, Writing - original draft, Writing - review & editing, Supervision, Project administration.

Acknowledgements

The authors would like to thank the HPC-Lab leader and PI behind the "Tensor-GPU" project, Prof. Anne C. Elster, for access to the Nvidia DGX-2 system used in the experiments performed as part of this paper. Additionally, the authors would like to thank the IDUN cluster at NTNU for the provision of additional computing resources.

References

[1] Novatnack J, Nishino K. Scale-dependent/invariant local 3d shape descriptors for fully automatic registration of multiple sets of range images. In: European conference on computer vision. Springer; 2008. p. 440–53.

[2] Malassiotis S, Strintzis MG. Snapshots: a novel local surface descriptor and matching algorithm for robust 3d surface alignment. IEEE Trans Pattern Anal Mach Intell 2007;29(7):1285–90.

[3] Yamany SM, Farag AA. Surface signatures: an orientation independent free–form surface representation scheme for the purpose of objects registration and matching. IEEE Trans Pattern Anal Mach Intell 2002;24(8):1105–20.

[4] Ovsjanikov M, Li W, Guibas L, Mitra NJ. Exploration of continuous variability in collections of 3d shapes. ACM Transactions on Graphics (TOG) 2011;30(4):33.

[5] Hu R, Fan L, Liu L. Co-segmentation of 3d shapes via subspace clustering. In: Computer graphics forum, 31. Wiley Online Library; 2012. p. 1703–13.

[6] Wu Z, Wang Y, Shou R, Chen B, Liu X. Unsupervised co-segmentation of 3d shapes via affinity aggregation spectral clustering. Computers & Graphics 2013;37(6):628–37.

[7] Feng C, Jalba AC, Telea AC. A Descriptor for Voxel Shapes Based on the Skeleton Cut Space. Eurographics Workshop on 3D Object Retrieval. Ferreira A, Giachetti A, Giorgi D, editors. The Eurographics Association; 2016. ISBN 978-3-03868-004-8. doi:10.2312/3dor.20161082.

[8] Craciun D, Levieux G, Montes M. Shape Similarity System driven by Digital Elevation Models for Non-rigid Shape Retrieval. Eurographics Workshop on 3D Object Retrieval. Pratikakis I, Dupont F, Ovsjanikov M, editors. The Eurographics Association; 2017. ISBN 978-3-03868-030-7. doi:10.2312/3dor.20171051.

[9] Guo Y, Bennamoun M, Sohel F, Lu M, Wan J, Kwok NM. A comprehensive performance evaluation of 3d local feature descriptors. Int J Comput Vis 2016;116(1):66–89.

[10] Johnson AE, Hebert M. Using spin images for efficient object recognition in cluttered 3d scenes. IEEE Trans Pattern Anal Mach Intell 1999;21(5):433–49.

[11] Huber DF, Hebert M. Fully automatic registration of multiple 3d data sets. Image Vis Comput 2003;21(7):637–50.

[12] Kakadiaris IA, Passalis G, Toderici G, Murtuza MN, Lu Y, Karampatziakis N, et al. Three-dimensional face recognition in the presence of facial expressions: an annotated deformable model approach. IEEE Trans Pattern Anal Mach Intell 2007;29(4):640–9.

[13] Carmichael O, Huber D, Hebert M. Large data sets and confusing scenes in 3-d surface matching and recognition. In: 3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on. IEEE; 1999. p. 358–67.

[14] Assfalg J, Bertini M, Del Bimbo A, Pala P. Content-based retrieval of 3-d objects using spin image signatures. IEEE Trans Multimedia 2007;9(3):589–99.

[15] Dinh HQ, Kropac S. Multi-resolution spin-images. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 1. IEEE; 2006. p. 863–70.

[16] Guo Y, Sohel FA, Bennamoun M, Lu M, Wan J. Trisi: A distinctive local surface descriptor for 3d modeling and object recognition.. In: GRAPP/IVAPP; 2013. p. 86–93.

[17] Davis N, Braunreiter D, Tebcherani C, Tanida M. 3d object matching on the gpu using spin-image surface matching. In: Advanced Signal Processing Algorithms, Architectures, and Implementations XVIII, 7074. International Society for Optics and Photonics; 2008. p. 707408.

[18] Gerlach AR, Walker BK. Accelerating robust 3d pose estimation utilizing a graphics processing unit. In: Intelligent Robots and Computer Vision XXVIII: Algorithms and Techniques, 7878. International Society for Optics and Photonics; 2011. p. 78780V.

[19] Liang L, Szymczak A, Wei M. Geodesic spin contour for partial near-isometric matching. Computers & Graphics 2015;46:156–71.

[20] Pasqualotto G, Zanuttigh P, Cortelazzo GM. Combining color and shape descriptors for 3d model retrieval. Signal Process Image Commun 2013;28(6):608–23.

[21] Frome A, Huber D, Kolluri R, Bülow T, Malik J. Recognizing objects in range data using regional point descriptors. In: European conference on computer vision. Springer; 2004. p. 224–37.

[22] Mian AS, Bennamoun M, Owens R. Three-dimensional model-based object recognition and segmentation in cluttered scenes. IEEE Trans Pattern Anal Mach Intell 2006;28(10):1584–601.

[23] Flint A, Dick A, Van Den Hengel A. Thrift: Local 3d structure recognition. In: 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007). IEEE; 2007. p. 182–8.

[24] Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vis 2004;60(2):91–110.

[25] Chen H, Bhanu B. 3D free-form object recognition in range images using local surface patches. Pattern Recognit Lett 2007;28(10):1252–62.

[26] Deng H, Birdal T, Ilic S. Ppfnet: Global context aware local features for robust 3d point matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 195–205.

[27] Zeng A, Song S, Nießner M, Fisher M, Xiao J, Funkhouser T. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. p. 1802–11.

[28] Ioannidou A, Chatzilari E, Nikolopoulos S, Kompatsiaris I. Deep learning advances in computer vision with 3d data: a survey. ACM Computing Surveys (CSUR) 2017;50(2):1–38.

[29] van Blokland B.I., Theoharis T., Elster A.C.. Quasi spin images2018;.

[30] Tombari F, Salti S, Di Stefano L. Unique signatures of histograms for local surface description. In: European conference on computer vision. Springer; 2010. p. 356–69.

[31] Guo Y, Sohel F, Bennamoun M, Lu M, Wan J. Rotational projection statistics for 3d local surface description and object recognition. Int J Comput Vis 2013;105(1):63–86.

[32] Rusu RB, Cousins S. 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA); 2011. Shanghai, China

[33] Savva M, Yu F, Su H, Kanezaki A, Furuya T, Ohbuchi R, et al. Shrec17 track large-scale 3d shape retrieval from shapenet core55. In: Proceedings of the Eurographics Workshop on 3D Object Retrieval; 2017.