Eurographics Workshop on 3D Object Retrieval (2018)
A. Telea, T. Theoharis, and R. C. Veltkamp (Editors)

# Microshapes: Efficient Querying of 3D Object Collections based on Local Shape

Bart Iver van Blokland[1], Theoharis Theoharis[1]

[1]Norwegian University of Science and Technology, Norway

**Abstract**

*Content-based querying of 3D object collections has the intrinsic difficulty of creating the query object and previous approaches have concentrated in producing global simplifications such as sketches. In contrast, in this paper, the concept of querying 3D object collections based on local shape is introduced.* Microshapes *are very promising in terms of generality and applicability and is based on a variation of the spin image descriptor. This descriptor uses intersection counts to determine the presence of boundaries in the support volume. These boundaries can be used to recognise local shape similarity. Queries based on this descriptor are general, easy to specify and robust to geometric clutter.*

## 1. Introduction

A large number of objects, in particular man-made ones, inherently exhibit smaller shapes which together define the appearance of the whole object. For instance, shelving units are commonly constructed using rectangular planks, which in turn can be considered to contain straight corners as well as flat surfaces of various sizes.

On a semantic level, it is possible to describe objects by combinations of such "Microshapes", examples of which include *circular*, *rounded corner*, *slight bend*, and *concave edge*.

Local shape descriptors which have been proposed to date primarily aim at creating a summary of their support volume [GBS*14] [GBS*16]. They exploit qualitative properties which are a *result* of microshapes. For instance, the curvature at a specific surface point on a model is the *result* of the shape being convex or concave at that location, rather than the cause.

In this paper, we present a novel method which is, amongst others, able to detect the occurrence of specific microshapes within an object, as defined by a 2D search query image which can be created intuitively and efficiently.

In addition to its matching capability, our method has some distinct advantages:

- Capable of matching occluded sections of a mesh
- Invariant to pose
- Resistant to geometric clutter
- Easily compressible and efficiently comparable due to its binary nature

## 2. Related Work

Of previously proposed approaches, those in the Bag-of-Visual-Words (BoVW) shape retrieval [CDF*04] category can be considered to be closest to Microshape images. These methods commonly compute a vector of descriptors from a sample object, which is in turn used to locate similar points or surfaces in other models. Several distinct approaches have been proposed in this paradigm which were shown to have state-of-the-art performance. These include using Global Fisher vectors [PST*15] and image features computed from panoramic views [STP13]. While our method can potentially be applied in the BoVW paradigm, we consider Microshape query construction sufficiently intuitive such that queries can be formulated and used for querying directly. Using a query model is therefore not a necessity. Moreover, Microshapes search for curves or lines, where existing methods have focused on matching surface patches.

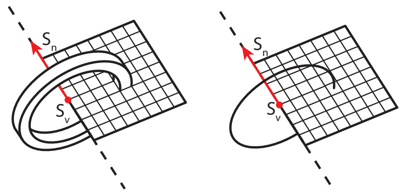
In terms of using curves for querying objects, sketch-based methods could to some extent be considered to use a similar approach to the one presented in this paper. A wide variety of such methods have previously been proposed, whose general goal is to match a user sketch of a desired object to objects in a database. This is commonly done by generating views of each mesh in the database from different angles, and comparing the shape of outlines and edges to those drawn in the sketched query through various means [SXY*11] [WKL15] [ERB*12]. However, these methods match entire sketches against entire models, and as such do not allow smaller shapes to be located.

The spin image, initially proposed by Johnson *et al.* [JH99], is a descriptor generated by rotating a square plane divided into pixels around a central axis for one revolution, and subsequently measuring the area of the mesh intersecting with the torus-like volume

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

DOI: 10.2312/3dor.20181047

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org




**(a)** **(b)**

**Figure 1:** *An illustration of the main difference between the original spin image, which measures the mesh area intersecting a torus-like volume generated for each pixel shown in 1a, and a quasi spin image, which measures the number of intersections made by a circle with the mesh surface for each pixel, as shown in 1b.*

generated by each pixel (as shown in Figure 1a). In the author's implementation, the area computation is approximated by accumulating uniformly sampled points instead. The descriptor has been shown to be noise resistant, and perform well in cluttered scenes. Moreover, due to the spin image's use of a cylindrical coordinate system centred around the spin vertex, it is pose invariant. This property is inherited by our method.

A number of methods have been derived from the original spin image in order to address various aspects of its weaknesses. Such methods include multi-resolution spin images, proposed by Dinh *et al*. [DK06], which aimed to address their lack of scale invariance, and computing signatures from spin images for faster matching, as proposed by Assfalg *et al*. [ABDBP07]. Additionally, a version of the spin image which does not use point samples to approximate area intersected per pixel was described by Carmichael *et al*. [CHH99].

A more recent method derived from the spin image, which bears some similarities to the one presented in this paper, is the Spin Contour proposed by Liuang *et al*. [LWS*16]. The spin contour is generated by computing a spin image, and subsequently locating the contour around all nonzero pixels on that image, which can be used for matching. However, the method only looks at the extreme outlines of a shape in cylindrical coordinate space, and, as we show in this paper, this discards curves present within the object which can potentially be used for matching. Moreover, the parts of a model which could be considered similar are not guaranteed to be part of the spin outline.

## 3. Background: Quasi Spin Images

The Microshape Image (MSI), introduced in this paper as an efficient query tool (see next Section), is a derivation of the Quasi Spin Image (QSI) descriptor, which was proposed as an efficient GPU-based alternative to Spin Images in [Ano]. We briefly describe the QSI below.

The QSI is constructed around a 3D point (referred to as the spin vertex), and a surface normal at this point (referred to as the spin normal). The spin vertex and spin normal combined describe a line, referred to as the central axis. Computing a QSI involves placing square plane (referred to as the spin plane) divided into pixels along the central axis, rotating it for one revolution, and in the process counting the number of intersections between each pixel centre and the mesh surface. A visualisation of this is shown in Figure 1b.

These intersections are effectively performed in a cylindrical coordinate space defined by the spin vertex and the central axis. Let $\alpha$ be a variable denoting the distance to the central axis and $\beta$ denote distance along the central axis (the spin normal) from the origin (the spin vertex). Imagine a sequence of planes $P$ placed perpendicular to the central axis at equal increments of $\beta$. Then, for each such plane, imagine concentric circles centred at the central axis with radii corresponding to linearly increasing values of $\alpha$. Each plane corresponds to a row of the QSI and each circle corresponds to a pixel within that row. The number of intersections of such a circle with the mesh surface gives the value of the corresponding QSI pixel. One aspect worth observing here is that the intersections of one of the above planes and the mesh surface produces two-dimensional curves.

Using intersection counts yields images which can both be computed consistently and are inherently free of noise. In contrast, the original spin image algorithm created a histogram of uniformly sampled point samples, and is therefore noisy. This is not acceptable in our case, as our method compares the values of neighbouring pixels (see next section). Moreover, because pixels in the original Spin Image represent sums of areas, distinguishing internal borders of objects is not possible. The properties exploited by our proposed descriptor therefore only exist in QSI images.

## 4. Microshape Images

The use of the MSI for querying 3D object collections based on local shape, was inspired by a specific observation in the behaviour of the QSI, which exploits its noise-free and consistency properties. Note that the intersection of the object mesh with the plane $P$ used when computing a QSI, produces planar curves, as shown in Figure 2. The key observation which the MSI image exploits, is that the number of intersections with these curves, and thus the mesh surface, encountered by circles at linearly increasing radii only decreases *whenever a specific section of the mesh is no longer encountered*.

Since the MSI contains only the changes in the intersection counts of the QSI, it represents local shape more directly and is therefore more suitable as a local shape descriptor.

A reduction in intersection count can be observed in the vast majority of cases, irrespective of the presence of clutter, as shown in Figure 3. The only exception is whenever from one radius to the next, a section of the mesh which is no longer encountered, is replaced by another. However, this requires some rather specific circumstances and was not found to be a significant problem in practice. We therefore consider the MSI image to be resistant, albeit not immune, to clutter.

Moreover, the radii at which intersection counts decrease tend to be relatively similar across neighbouring rows of MSI pixels, describing shapes which are present in the object. An example of this is shown in Figure 4.

A database of microshape images corresponding to each vertex of a collection of 3D objects is queried by computing their distances

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

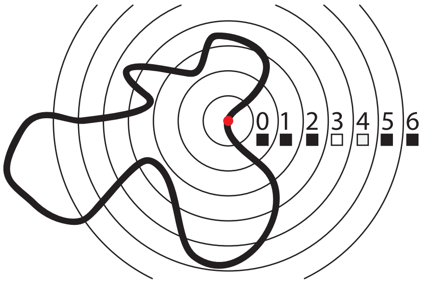



**Figure 2:** *The generation of a single row in the MSI image. The portion of a shape intersecting with a plane described by a point along the central axis and the spin normal is shown. Circles with linearly increasing radii are intersected with the mesh. The computed values of the corresponding row of pixels in the MSI image are shown below each row index.*

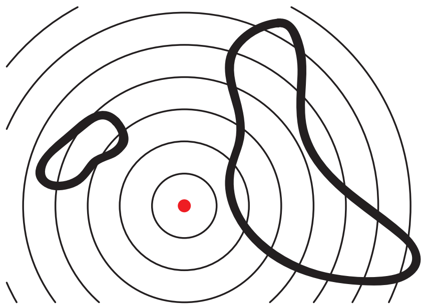


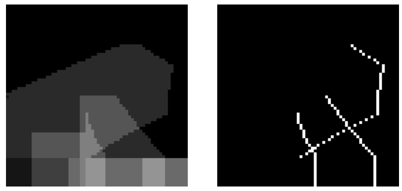
**Figure 3:** *The object on the left causes a decrease in the number of encountered intersections with increasing circle radii, irrespective of the presence of the object on the right.*

from a constructed query MSI. We will refer to the query MSI as the "needle image" and to a database MSI as a "haystack image". Algorithm 1 describes the Offline and Online parts of the proposed query method.

**Offline**
**for** *every 3D object O in haystack* **do**
  **for** *every vertex v in O* **do**
   - Compute QSI for v on GPU (Section 3)
   - Compute microshape image m from QSI (Section 4.1)
   - Store m as vertex v data
  **end**
**end**
**Online**
- Design needle (query) microshape image q
- Compute distance function between q and the microshape image of every vertex of every object in the collection (Section 4.2)
- Produce ranked retrieval list of 3D objects based on distances of vertices from q

**Algorithm 1:** Microshapes: offline and online algorithms

We have identified two limitations of our method. First, as MSI inherits some of the properties of QSI, the MSI are not scale invariant. However, the distance function is lenient and allows a certain degree of scale variations. Second, since MSI are discrete images and thus susceptible to aliasing effects, sampling detailed geometry which exhibit rapidly varying intersection counts (high frequencies), details can potentially be missed. But it should be noted that since MSI are bit vectors, they require 16 times less storage space compared to QSI, one can afford to generate them at a higher spatial resolution.



(a) (b)

**Figure 4:** *A sample quasi spin image (QSI) (4a) with its corresponding microshape image (MSI) (4b).*

### 4.1. Microshape Computation

Computing a microshape image (MSI) consists of two steps. First, for a given vertex (spin vertex) and its associated normal (spin normal), a QSI is generated. Then, the QSI is converted into a binary MSI as follows. The value of each pixel in the QSI (number of intersections) is compared to its right neighbour. If the number of intersections in the right pixel is smaller than those in the left pixel, the left pixel is set to "active" (1), and "inactive" (0) otherwise.

A sample QSI and its corresponding MSI is shown in Figure 4.

### 4.2. Microshape Distance Function

The main idea behind our distance function is to evaluate to which degree the pattern in needle MSI is contained in a haystack MSI. We have identified four properties a distance function should exhibit to achieve our goal, and our implementation satisfies these requirements.

First, if active pixels in the haystack MSI are close to active pixels in the needle MSI then a low distance value must be produced. Second, the distance function should allow for minor differences in active pixels to occur; this is to allow for rounding errors and minute differences in shape. Third, in order to avoid taking geometric clutter into account, we should *only* be concerned with the haystack MSI patterns that correspond to active patterns in the needle MSI. Fourth, we should avoid matches resulting from haystack MSI geometric clutter close to the needle MSI pattern.

The above requirements resulted in the following distance algorithm, which works on a row by row basis. It is worth mentioning here that because MSIs are binary, our implementation stores and processes them as bit vectors.

For each row, we look at the location of the active bits in the needle MSI, and for each of these bits locate the closest active bit in the same row of the haystack MSI; this covers the first and second requirements. To satisfy the third requirement, we seek haystack

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

MSI matches in a width of 2 pixels from the active needle pixels; unmatched haystack active pixels are given the maximum penalty of 2, since they have not been matched within the 2-pixel band. To address the fourth requirement, we calculate the hamming distance between needle and haystack MSIs within the 2-pixel band around each active needle pixel and add this to the distance measure of the row.

Our implementation uses 64x64 images, which means an entire row of an MSI can be stored in a single unsigned 64-bit integer. Our comparison method thus uses boolean bitwise operations, which allows it to be implemented efficiently.

The distance algorithm can be implemented efficiently by taking the bit string representing a needle image row and iteratively filtering away bits for which corresponding bits are located in the haystack row at increasing distances (up to 2 pixels in each direction). This can be done through a series of bit shifts, boolean operators, and the population count instruction (which counts the number of set bits in a bit string).

## 5. Experiments

We evaluated the potential of our method by designing query MSI images to retrieve objects from the SHREC17 training dataset [SYS*17] based on the local shape characteristics described by the MSI.

We generated an MSI for each vertex/normal present in each model, and created a database of the produced MSIs. The size of the spin plane was set to be half of the side of a cube whose volume is equivalent to the axis-aligned bounding box of the input model. This implies that MSI are generated at a scale close to that of the model. The resolution of the MSI was set to 64x64 pixels.

For each query MSI, we computed a distance score against all vertex MSIs and sorted the corresponding objects by ascending score. In Figures 5 to 8, we have indicated the location of the detected vertex with a red dot, and any matched microshapes with red lines in example objects from the top ranks of the retrieval lists of particular interest. The top 20 unique objects are shown in Figures 9, 10, 11, and 12.

It should be noted that our search algorithm matches individual vertices. Some objects contain repeating shapes or patterns, and will therefore often appear multiple times in the retrieval list. Only the top appearance of each object has been indicated in the results list.

The generality of the approach should be observed here; simply giving the main characteristic of a local shape results in objects that possess it from various distinct object classes. This is a complex endeavour with global content-based queries. Figure 6a indicates that queries can be formulated which do not require the desired local shape to be originating from the sampled vertex.

It is particularly worth mentioning that the query MSIs were created in a simple image editor in the order of 1 minute each.

## 6. Conclusion and Future Work

We have strong indications that querying based on microshapes is very general, widely applicable and quite simple in terms of the user interface required. Its local nature appears to simplify the problem of query formulation when searching 3D object collections based on content. A descriptor derived from a variant of the spin image that can be effectively computed on the GPU proved a highly suitable microshape descriptor.

While initial qualitative experiments indicate that the microshape method is very promising, a more thorough evaluation is required leading to quantitative results. For example, we can annotate the 3D objects of a certain collection with the microshapes (out of a finite set) that each contains and then perform microshape queries and count false positives and false negatives, thus leading to the standard retrieval metrics.

We believe the work presented in this paper opens the possibility for semantically querying 3D object collections, based on individual local features of a desired object. This is in contrast to conventional querying approaches, which concentrate on global appearance and thus do not offer the possibility of specifying detailed desired local shape characteristics.

## References

[ABDBP07] Assfalg J., Bertini M., Del Bimbo A., Pala P.: Content-based retrieval of 3-d objects using spin image signatures. *IEEE Transactions on Multimedia 9*, 3 (2007), 589–599. 2

[Ano] Anonymous: Efficient quasi spin image generation on the gpu. *submitted for publication*. (Authors have been anonimised for peer review). 2

[CDF*04] Csurka G., Dance C., Fan L., Willamowski J., Bray C.: Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV* (2004), vol. 1, Prague, pp. 1–2. 1

[CHH99] 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), pp. 358–367. doi:10.1109/IM.1999.805366. 2

[DK06] Dinh H. Q., Kropac S.: Multi-resolution spin-images. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (2006), vol. 1, IEEE, pp. 863–870. 2

[ERB*12] Eitz M., Richter R., Boubekeur T., Hildebrand K., Alexa M.: Sketch-based shape retrieval. *ACM Trans. Graph. 31*, 4 (2012), 31–1. 1

[GBS*14] Guo Y., Bennamoun M., Sohel F., Lu M., Wan J.: 3d object recognition in cluttered scenes with local surface features: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence 36*, 11 (2014), 2270–2287. 1

[GBS*16] Guo Y., Bennamoun M., Sohel F., Lu M., Wan J., Kwok N. M.: A comprehensive performance evaluation of 3d local feature descriptors. *International Journal of Computer Vision 116*, 1 (2016), 66–89. 1

[JH99] Johnson A. E., Hebert M.: Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence 21*, 5 (1999), 433–449. 1

[LWS*16] Liang L., Wei M., Szymczak A., Pang W.-M., Wang M.: Spin contour. *IEEE Transactions on Multimedia 18*, 11 (2016), 2282–2292. 2

[PST*15] Pratikakis I., Spagnuolo M., Theoharis T., Van Gool L., Veltkamp R.: Partial 3d object retrieval combining local shape descriptors with global fisher vectors. 1

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

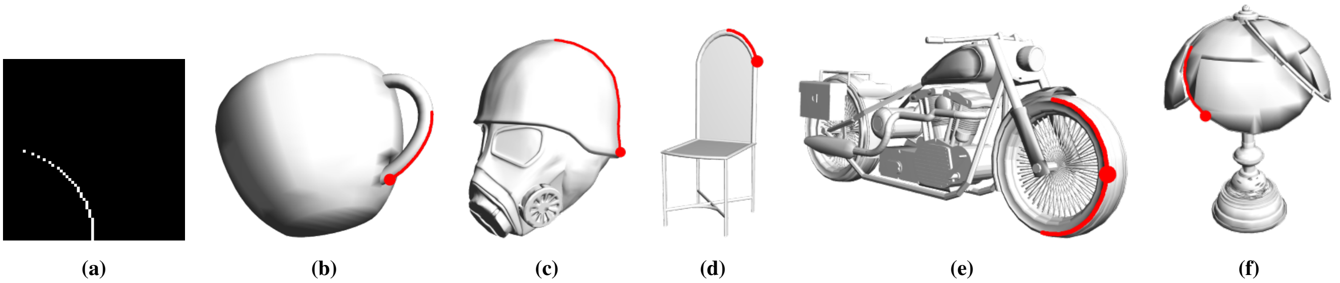


**Figure 5:** *Search results produced by a query for a perfect quarter circle, as shown in 5a. The meshes shown in 5b to 5d are example objects from the top ranks of the retrieval list.*

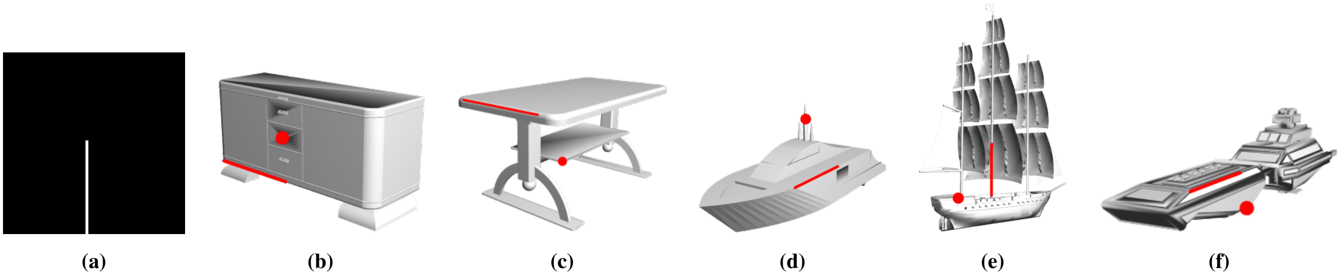


**Figure 6:** *Search results returned by our algorithm based on the query MSI image shown in Figure 6a, representing a long straight edge some distance away from the sample point.*

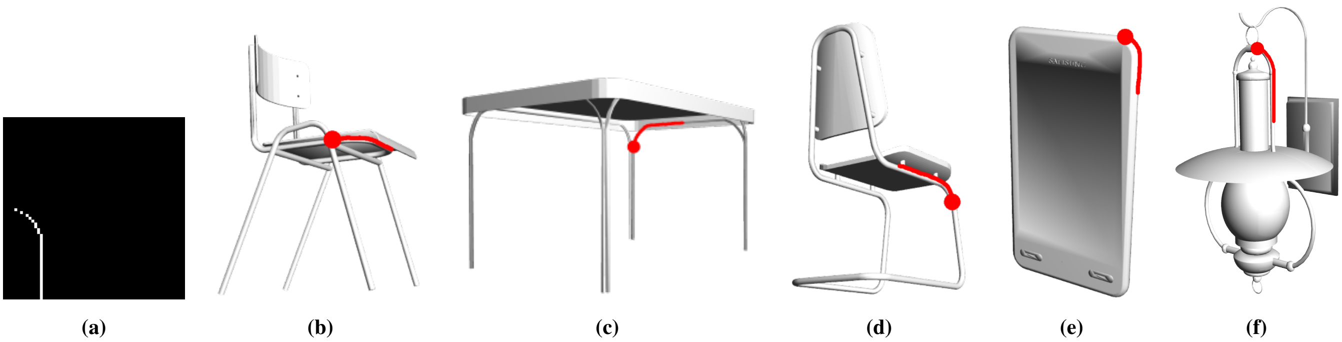


**Figure 7:** *Search results returned by our algorithm based on the query MSI image shown in Figure 7a, representing a rounded corner followed by a straight edge.*

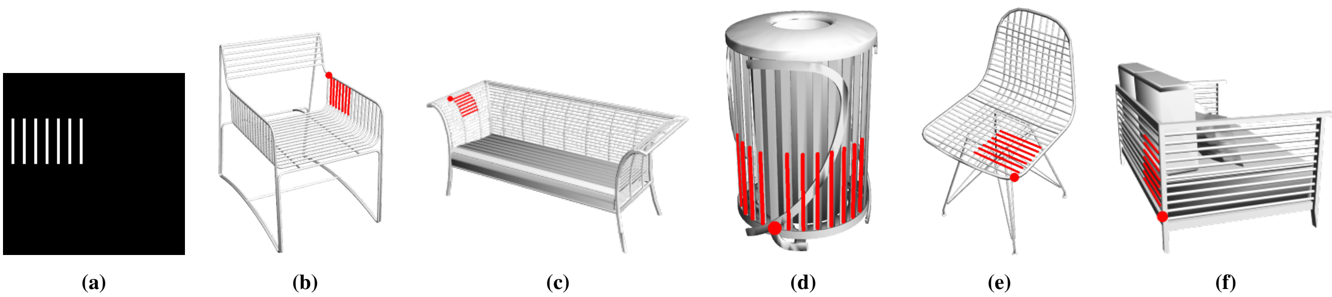


**Figure 8:** *Search results returned by our algorithm based on the query MSI image shown in Figure 8a. The image represents a "grating"-like pattern.*

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

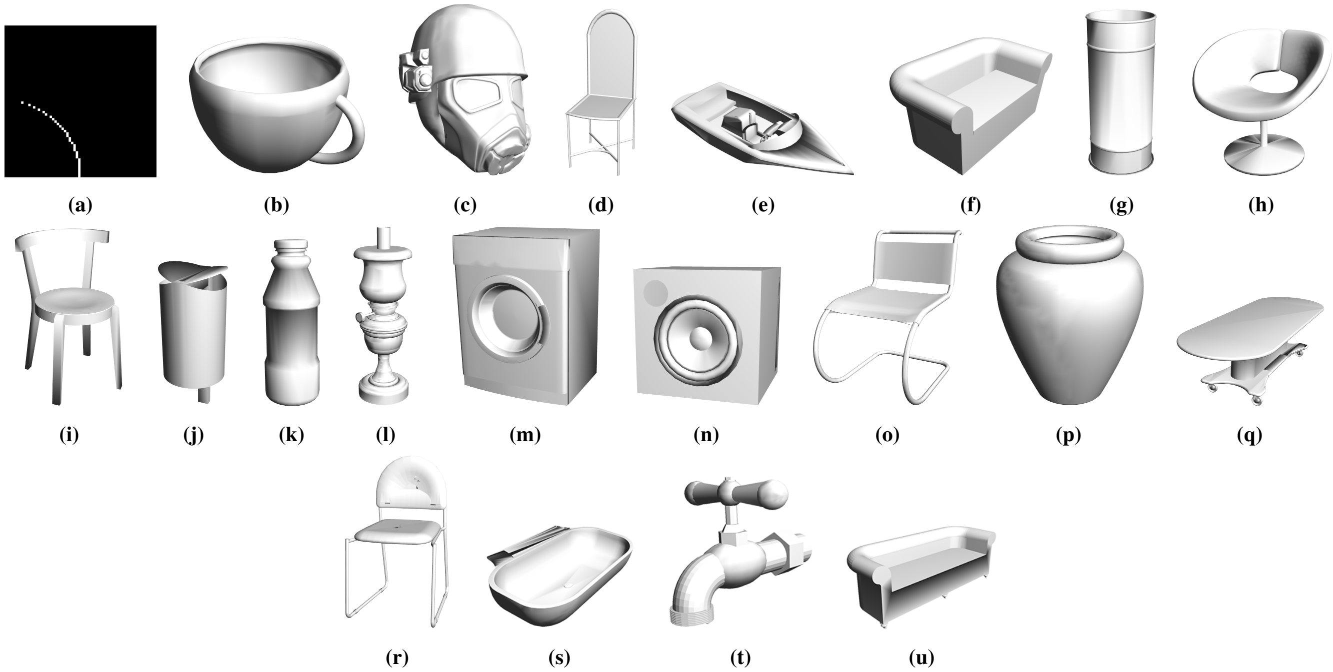



**Figure 9:** *The top 20 unique models (shown in order) in the search results returned by our algorithm for the "quarter circle" query image shown in 9a.*

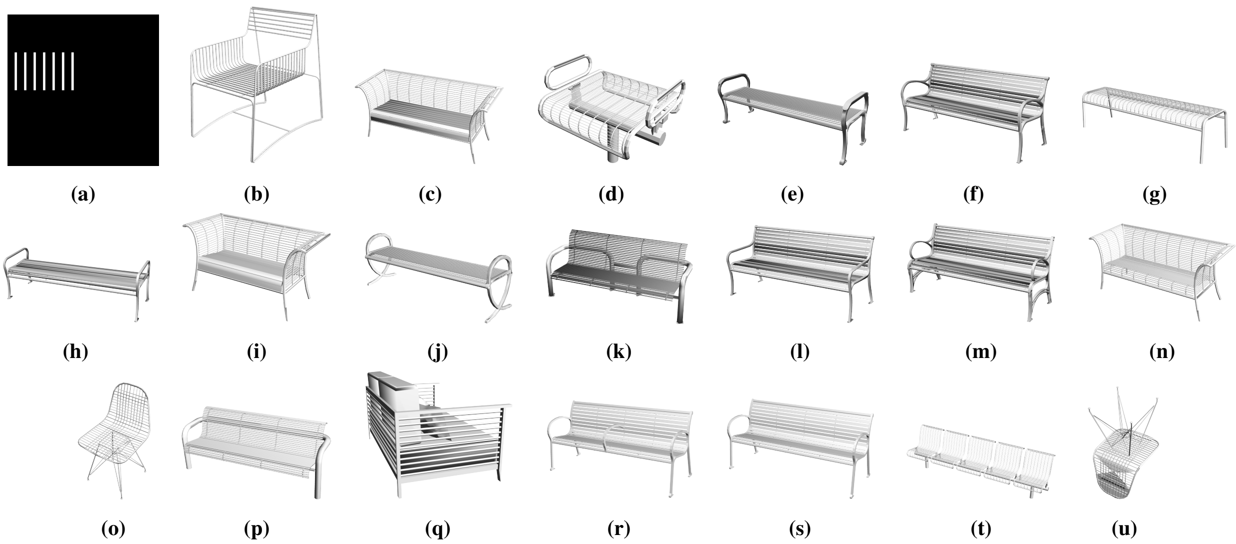



**Figure 10:** *The top 20 unique models (shown in order) in the search results returned by our algorithm for the "grating" query image shown in 10a.*

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

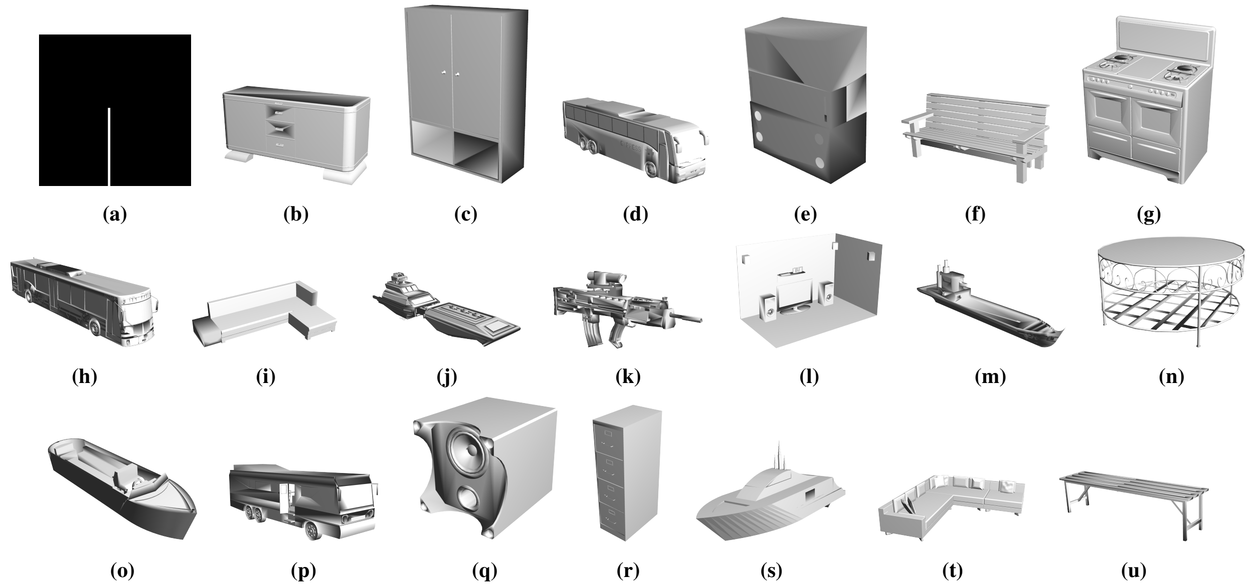



**Figure 11:** *The top 20 unique models (shown in order) in the search results returned by our algorithm for the "long straight edge" query image shown in 11a.*


**Figure 12:** *The top 20 unique models (shown in order) in the search results returned by our algorithm for the "rounded corner" query image shown in 12a.*

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

[STP13] SFIKAS K., THEOHARIS T., PRATIKAKIS I.: 3d object retrieval via range image queries in a bag-of-visual-words context. *The Visual Computer 29*, 12 (2013), 1351–1361. 1

[SXY*11] SHAO T., XU W., YIN K., WANG J., ZHOU K., GUO B.: Discriminative sketch-based 3d model retrieval via robust shape matching. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library. 1

[SYS*17] SAVVA M., YU F., SU H., KANEZAKI A., FURUYA T., OHBUCHI R., ZHOU Z., YU R., BAI S., BAI X., AONO M., TATSUMA A., THERMOS S., AXENOPOULOS A., PAPADOPOULOS G. T., DARAS P., DENG X., ZHOUHUI L., LI B., JOHAN H., LU Y., MK S.: ShrecâĂŹ17 track large-scale 3d shape retrieval from shapenet core55. In *Proceedings of the Eurographics Workshop on 3D Object Retrieval* (2017). 4

[WKL15] WANG F., KANG L., LI Y.: Sketch-based 3d shape retrieval using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on* (2015), IEEE, pp. 1875–1883. 1

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.