Eurographics Workshop on 3D Object Retrieval (2016)
A. Ferreira, A. Giachetti, and D. Giorgi (Editors)

A descriptor for voxel shapes based on the skeleton cut space

C. Feng[1] and A. C. Jalba[2] and A. C. Telea[1]

[1]Institute Johann Bernoulli, University of Groningen, the Netherlands
[2]Dept. of Mathematics and Computer Science, TU Eindhoven, the Netherlands

Abstract

Two-dimensional medial axes and three-dimensional curve skeletons have been long used for shape retrieval tasks. In contrast, and despite their ability to fully capture shape geometry and topology, three-dimensional surface skeletons have seen much less usage in this context. We present here a framework for shape matching and retrieval based on such surface skeletons. To this end, we construct a space of cuts generated by the surface skeleton, which has desirable invariance properties with respect to shape size, rotation, translation, pose, and noise. Next, we extract a histogram-based descriptor from this cut space, and discuss three different metrics to compare such histograms for shape retrieval. We illustrate our proposal by showing our descriptor's effectiveness in shape retrieval using a known shape-database benchmark.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computation Geometry and Object Modeling—Curve, surface, solid, and object representations, I.4.7 [Image Processing and Computer Vision]: Feature Measurement—Size and shapes, I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Shape

1. Introduction

The increase in computational resources and scanning technologies has made large collections of 3D shapes readily available. With this increase has come the demand for methods and techniques for searching for specific shapes in such databases [SHP97, FMKC03]. Within this field, *content-based shape retrieval* (CBSR) focuses on efficiently finding the most similar shapes to a given example shape from a given shape collection. Many classes of CBSR methods exist, based on various techniques, such as parametric templates [GF15], descriptor-based methods [JTV08], and skeleton-based methods [SSGD03, XHS08].

Among the above, skeleton-based methods have the important advantage as being able to compare shapes at a high level, and based not only on their geometry, but also, specifically, their topology. This supports CBSR applications where the search is driven by *global* shape properties, such as structure and topology [SSGD03, CDS*05], in addition to the more commonly used *local* geometry and texture properties. Classical medial axes (in 2D) and curve skeletons (in 3D) have proven here to be efficient and effective descriptors [SKK04, SSGD03]. In the 3D case, however, curve skeletons can capture well shape topology, and partially geometry, only for locally quasi-tubular shapes [CSM07]. In contrast, surface skeletons capture well both topology and geometry for any 3D shape [JKT13]. However, to our knowledge, such skeletons have not yet been used for CBSR.

We present in this paper a shape descriptor that uses surface skeletons of 3D voxel shapes for CBSR. From these skeletons, we construct a so-called cut space that describes the local shape thickness properties in a multiscale way, *i.e.*, ignores small details which are less relevant for CBSR, but captures the actual shape thickness well. We next reduce this cut space to a histogram descriptor that efficiently and effectively captures thickness properties, and is also invariant to isometric shape transformations, pose, noise, and sampling resolution. We next study two existing metrics, and propose two new metrics, to compare such histograms for CBSR. We demonstrate our proposal by applying it to a well-known 3D shape benchmark database. Summarizing, the main contribution of this paper is showing that surface skeletons have the potential to be efficient and effective instruments for constructing compact thickness-descriptors that, next, perform well for CBSR applications.

This paper is structured as follows: Section 2 reviews related work, with a focus on skeleton-based CBSR. Section 3 presents our method. Section 4 illustrates our method on a variety of 3D shapes. Section 5 discusses our method. Section 6 concludes the paper.

2. Related Work

Many methods have been proposed for CBSR [JTV08]. Most such methods use a two-step approach: First, a so-called descriptor is computed from a shape, aiming to compactly capture the relevant shape properties for the search process, and also to be resistant to shape changes deemed as irrelevant for the same context, such as isometric transformations and noise [KFR03]. Next, the best-match for the given descriptor is searched for in a database of shapes, which is organized (indexed) so as to speed up the search process.

DOI: 10.2312/3dor.20161082

Our focus in this work is the first above step – descriptor computation. To this end, we next review typical descriptors used in CBSR, and next focus on skeletal descriptors, which are central, and specific, to our method.

Shape descriptors for CBSR: Osada *et al.* [OFCD02] describe a shape by a distribution capturing its geometric properties such as angle, distance, and volume between random point-pairs on the shape surface. Kazhdan *et al.* [KFR03] propose a method to create a descriptor which is rotation invariant, based on spherical harmonics. The descriptor is further refined in [KFR04] to capture a shape's reflective and rotational symmetry. However, no exact accuracy information is provided, except for a stated 15% improvement in precision-recall plots as compared to PCA alignment of models prior to retrieval. The spherical extent function (SEF) measures the thickness of a shape along rays passing through its origin [SV01]. While simple, this descriptor does not work well for thin and entangled shapes.

Liu *et al* [LKyS03] propose the directional histogram model (DHM) that is very similar in spirit with, and also shares the limitations of, the SEF. Recently, Schmidt *et al.* [SST*15] extended the DHM to a two-dimensional descriptor combining shape thickness and depth complexity (number of intersections of a ray with the shape), thereby capturing both shape geometry and topology. Both above methods, however, require thousands of 2D views of the shape to be rendered and analyzed, which becomes expensive even when using GPU acceleration.

Jain and Zhang [JZ07] propose to represent each shape by the eigenvectors of a certain affinity matrix constructed such that normalization against rigid-body transformations, uniform scaling and bending is achieved. They also show that the retrieval performance of the light-field [CTSO03] and spherical-harmonic [KFR03] descriptors is improved by 5...10% when applied on the proposed spectral embedding. Other state-of-the-art methods include covariance [TL15] and diffusion [BBK*10, SOG09, FXD*15] -based descriptors.

Apart from the above descriptors that use only shape geometry, other descriptors use additional data such as texture and lighting. For instance, the DB-VLAT descriptor proposed in [TKA12] combines dense SIFT descriptors taken from several 2D shape views. While showing very good precision and recall, this descriptor requires careful prior shape alignment, and appears to be very computationally intensive. More importantly, texture and/or lighting data is not always available, such as in the case of raw 3D voxel shapes.

Finally, machine learning techniques such as the bag-of-features and deep neural networks (DNNs) have also been used to learn shape descriptors, thus improving their retrieval performance, see, *e.g.*, [LGS10, FXD*15] and references therein. However, training DNNs can be challenging, as this requires numerous examples to learn from.

Medial descriptors: Skeletons, also known as medial axes, were introduced by Blum for the 2D case [Blu67]. For the 2D case, medial axes jointly capture a shape's geometry and topology by the so-called medial axis transform, which is an exact dual representation of a shape as compared to a standard boundary representation (see also Sec. 3.1). Siddiqi *et al.* [SSDZ99] define the shock graph extracted from the shock grammar [SK96] and match such graphs to find similar 2D shapes. However, while this method can effectively capture topology, it is less good in capturing geometry, and was not extended to 3D shapes. Sebastian *et al.* [SKK04] also use shock graphs to recognize shapes, based on a relatively expensive edit-type distance metric (minimum amount of changes needed to modify a shock graph into another one). However, this method can only handle 2D shapes. Similarly, Xie *et al.* compare shapes by segmenting them using their skeletons' junctions and performing a part-by-part matching [XHS08]. As [BL08], this method can only handle 2D shapes.

In 3D, the by far most used medial descriptors are *curve* skeletons, which essentially capture a shape's local circular symmetry, and are relatively easy to compute [CSM07, SJT14]. As such, several methods have used curve skeletons for CBSR. Sundar *et al.* [SSGD03] use 3D curve skeletons to match shapes by extracting individual skeletal branches to construct a graph, and next using graph-matching algorithms to compare such shape graphs. However, this method is quite complex and computationally expensive. The method is next refined in [CDS*05] to use the Earth Mover's Distance (EMD) technique to compare 3D curve skeletons to match shapes [RTG98]. For 2D shapes, Xiang *et al.* [BL08] propose an alternative to [SSGD03] by first pruning 2D medial axes to remove irrelevant branches (expensive for matching) using the DCE method [BLL07], and next using paths between skeletal endpoints to construct a graph. While faster than [SSGD03], the method cannot effectively handle shapes having few salient skeletal endpoints, *i.e.*, shapes which are not well described by curve skeletons, and, as mentioned, can only treat 2D shapes.

Apart from curve skeletons, 3D shapes admit also *surface* skeletons, which fully capture their geometry and topology regardless of the shape type and are effectively dual representations of shape boundary descriptions [SJT14]. However, efficiently computing surface skeletons for complex 3D shapes has been considerably harder than computing curve skeletons, due to the higher complexity of such descriptors, but also to their well-known sensitivity to small shape- surface perturbations. Recent algorithms have changed this, allowing the near-real-time computation of surface skeletons both from mesh-based models [JKT13] and voxel-based models [JST15]. Key to their usability, such methods also regularize the produced surface skeletons by efficiently computing so-called multiscale importance metrics, which ensure that only large surface details will generate skeletal branches [DS06, RvWT08a]. Summarizing the above, 3D curve skeletons have proved to be useful and efficient descriptors for CBSR, but are limited to mainly tubular shapes. Surface skeletons overcome the descriptive power of curve skeletons for any 3D shape type, but have only recently shown to be efficiently and robustly computable for complex 3D shapes. As such, the main aim of this paper is to show that recent advances in 3D surface skeleton computation make these descriptors more efficient, more effective, and simpler to implement tools for CBSR than the well-known curve skeletons used for the same task.

3. Method

We next explain our CBSR descriptor proposal. Section 3.1 outlines relevant skeleton definitions. Section 3.2 outlines the construction of

the so-called 'cut space' which we use next to build our histogram-based shape descriptors (Sec. 3.3). Finally, Section 3.4 shows how such histogram descriptors can be efficiently and effectively compared. Figure 1 illustrates the proposed CBSR pipeline.

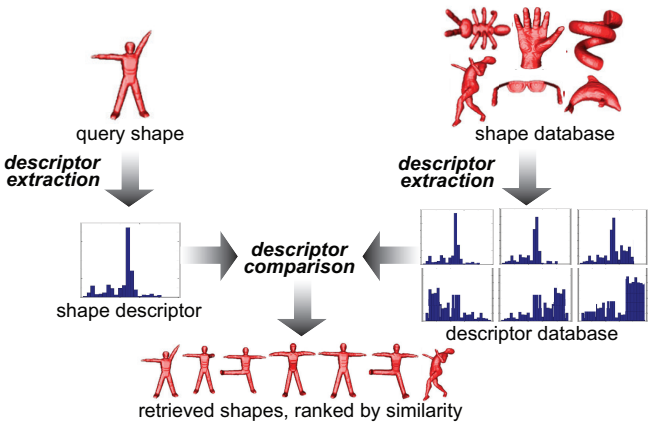

Figure 1: *Proposed CBSR pipeline. Our focus is the descriptor extraction and comparison (marked in bold).*

3.1. 3D Skeletonization

Let the Euclidean distance transform $DT_{\partial\Omega} : \Omega \to \mathbb{R}_+$ of a binary voxel shape $\Omega \subset \mathbb{Z}^3$ with boundary $\partial\Omega$ be denoted by

$$DT_{\partial\Omega}(\mathbf{x} \in \Omega) = \min_{\mathbf{y}\in\partial\Omega} \|\mathbf{x}-\mathbf{y}\|. \quad (1)$$

The medial surface, or surface skeleton, of $\partial\Omega$ is next defined as

$$S_{\partial\Omega} = \{\mathbf{x} \in \Omega | \exists \{\mathbf{f}_1, \mathbf{f}_2\} \subset \partial\Omega, \mathbf{f}_1 \neq \mathbf{f}_2, \\ \|\mathbf{x}-\mathbf{f}_1\| = \|\mathbf{x}-\mathbf{f}_2\| = DT_{\partial\Omega}(\mathbf{x})\} \quad (2)$$

where $\mathbf{f}_1$ and $\mathbf{f}_2$ are the contact, or feature, points with $\partial\Omega$ of the maximally inscribed ball in Ω centered at $\mathbf{x}$ [GK04, RH08]. These define the feature transform $FT_{\partial\Omega} : \Omega \to \mathcal{P}(\partial\Omega)$

$$FT_{\partial\Omega}(\mathbf{x} \in \Omega) = \operatorname*{argmin}_{\mathbf{y}\in\partial\Omega} \|\mathbf{x}-\mathbf{y}\|. \quad (3)$$

The medial surface implied by Eqn. 2 consists of a complex set of 2D manifolds embedded in 3D. Their direct computation, following Eqn. 2, is sensitive to small-scale noise on $\partial\Omega$, especially when using voxel-based discretizations of Ω. To alleviate this, $S_{\partial\Omega}$ can be regularized by a computing a metric $\rho : S_{\partial\Omega} \to \mathbb{R}_+$ such as the medial geodesic function (MGF) which sets $\rho(\mathbf{x})$ to the length of the shortest path on $\partial\Omega$ between the two feature points of $\mathbf{x}$ [DS06]. As the MGF monotonically increases from the medial surface boundary to its center, upper thresholding it always yields connected and noise-free simplified medial surfaces [RvWT08b, JKT13]. A similar regularization metric, which is faster to compute than the MGF, is proposed in [JST15] based on a mass advection process from $\partial\Omega$ onto $S_{\partial\Omega}$. In our work next, we will use the regularized surface skeletons produced by [JST15].

3.2. Cut space construction

The first step of our method is to construct a so-called *cut space* from the surface skeleton $S_{\partial\Omega}$, following the process proposed in [FJT15] (see Fig. 2a-e, for a hand model): For each point $\mathbf{x} \in S_{\partial\Omega}$, we construct a closed curve $C(\mathbf{x}) \subset \partial\Omega$ as the union $\gamma_1 \cup \gamma_2 \cup \gamma_3$, where γ_1 is the shortest path on $\partial\Omega$ between the two feature points $\mathbf{f}_1$ and $\mathbf{f}_2$ of $\mathbf{x}$ (Fig. 2a); and γ_2 and γ_3 are the shortest-paths on $\partial\Omega$ between $\mathbf{o}$, the opposite point on $\partial\Omega$ of the midpoint $\mathbf{m}$ of γ_1 with respect to $\mathbf{x}$, and $\mathbf{f}_1$ and $\mathbf{f}_2$, respectively ((Fig. 2b-d). For full implementation details, we refer to [FJT15].

3.3. Cut thickness histogram

Key to our shape descriptor idea is that the length $\|C(\mathbf{x})\|$ of a cut $C(\mathbf{x})$ is a good local descriptor of the shape's properties around point $\mathbf{x}$. We argue this by following the analysis in [FJT15]: The cut-space $CS = \bigcup_{\mathbf{x}\in S_{\partial\Omega}} C(\mathbf{x})$ captures, locally, the shape's symmetry and thickness in an effective and natural way. That is, CS contains all cuts which are locally smooth, tight-wrapping around $\partial\Omega$, and oriented orthogonal to Ω's local symmetry axis. As these cuts have proven to be good for shape segmentation [FJT15], we argue that they are also good for describing local shape thickness. Fig. 2e shows the cut space CS for our hand model, colored to emphasize the difference between short and long cuts. For all implementation details of the construction of CS, we refer further to [FJT15].

As mentioned in [FJT15], CS contains a small number of cuts which are not orthogonal to the local symmetry axis, or curve skeleton, of Ω. Upon closer examination, we noticed that these correspond to very short geodesics γ_1, *i.e.*, skeleton points $\mathbf{x}$ whose two feature points $\mathbf{f}_1$ and $\mathbf{f}_2$ are very close on $\partial\Omega$. The presence of such cuts unnecessarily perturbs the computation of $H(\Omega)$. This problem is well known in 3D skeletonization: Skeleton points having small feature-vector angles tend to be very unstable [FLM03]. Hence, we remove from CS all cuts $C(\mathbf{x})$ for which $\|\gamma_1\|/\|C(\mathbf{x})\| < 0.3$, an empirically determined value which gave good results in all our tests. The resulting regularized cut-space CS^r is next used to robustly compute our shape descriptor.

We stress that the local shape-thickness estimation $\|C(\mathbf{x})\|$ adapts itself to both the shape *orientation* (C is orthogonal to Ω's local symmetry axis, or curve skeleton, of Ω) and the shape's local *geometry* (C is piecewise geodesic by construction, thus smooth). This is in high contrast to [SST*15, LKyS03], who estimate the local shape thickness in *arbitrary* directions, *i.e.*, regardless of the shape's curve-skeleton orientation. The above papers recognize that this can yield significant thickness-estimation noise. Indeed, formally, just one single direction (orthogonal to Ω's curve-skeleton) yields the 'true' shape thickness, as this thickness value is unique.

Given our cut space CS^r, we next build a histogram $H(\Omega)$ of all cut lengths $\{\|C(\mathbf{x})\| | \forall \mathbf{C} \in CS^r\}$. Since $S_{\partial\Omega}$ is rotation and translation invariant with respect to Ω, so is CS^r, and thus so is $H(\Omega)$. To achieve scaling invariance, we normalize the cut lengths by their median value over CS^r. This essentially makes cut lengths relative to the shape size, and thus makes next $H(\Omega)$ invariant to the scale (size) of Ω. To make H independent on the number of cuts (which depends in turn on the voxelization resolution of Ω and $S_{\partial\Omega}$), we normalize the bin-values $H(\Omega)^i$ by the total cut count $\|CS^r\|$. Using $B = 20$ bins $H(\Omega)^i$, $1 \leq i \leq B$, for constructing $H(\Omega)$ proved to be a very good balance between accuracy and level-of-detail.

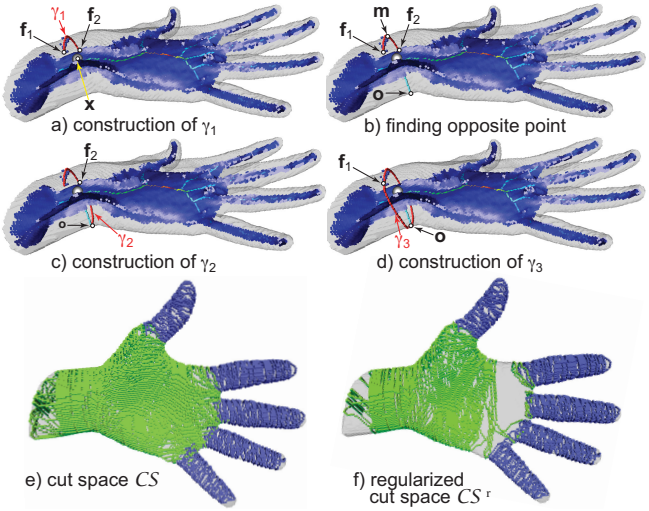

Figure 2: *Construction of regularized cut space CS^r.*

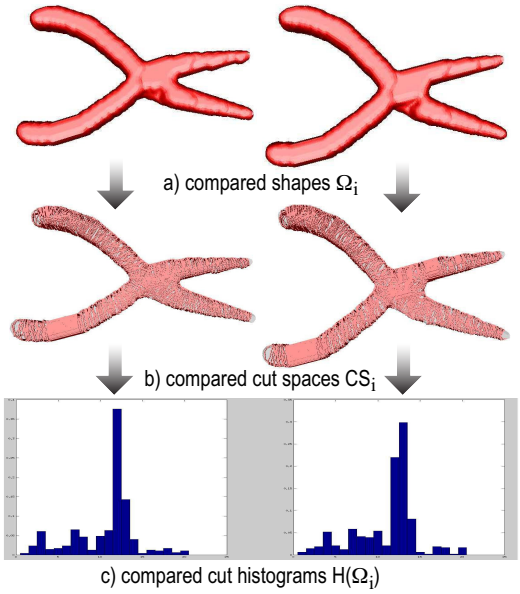

Figure 3: *Two similar shapes and their cut-spaces and histograms.*

3.4. Shape matching distances

Given two shapes Ω_1 and Ω_2 and their respective histograms $H(\Omega_1)$ and $H(\Omega_2)$, denoted next H_1 and H_2 for brevity, we compare these by using a distance function $d(H_1,H_2) \in [0,1]$. To this end, we studied four such distances, as follows.

Hellinger distance: We first consider the simple Hellinger distance $d_H(H_1,H_2) = \sum_{i=1}^{B} \left(\sqrt{H_1^i} - \sqrt{H_2^i}\right)^2$. Although d_H is not an optimal metric to compare shapes, as we will also show next, it has been used in previous CBSR applications [SST*15, JTV08], so we include it here for completeness. Upon close examination, we noticed that d_H causes false negatives, *i.e.*, high distance values between two visually very similar shapes. Figure 3 shows such an example, where $d_H = 0.575$ for two quite similar shapes. We found similar problems when using other simple histogram-distances, such as the L_1 norm, Chi-square, and correlation metrics.

This problem can be understood if we consider the *coarse* nature of our cut-space: CS^r contains at most as many cuts as the size $\|S_{\partial\Omega}\|$ of the surface skeleton, which is bounded by the voxelization-resolution of our input shapes Ω, and next reduced by regularization of CS to CS^r. In practice, for a shape voxelized at 300^3 resolution, CS^r contains a few thousand cuts. In contrast, the thickness estimation in [SST*15] uses 500K thickness samples, which is *over three orders of mangitude* more. Similar high sampling resolutions are used in [LKyS03, SV01]. While super-sampling alleviates the false-negative problem, it is also very expensive. Hence, one key challenge for our method is how to increase the *robustness* of comparing histograms without increasing the voxelization resolution, and thus *cost*, of our pipeline. Apart from reducing computational costs, this will also allow using our method on coarsely-sampled shapes.

EMD: Another popular way to compare histograms is the Earth movers distance (EMD) [RTG98]. When comparing two histograms, this metric treats one histogram H_1 as mass and the second one H_2 as a container. The least amount of mass required to be moved from H_1 to H_2 to make the two histograms identical gives the distance $d_{EMD}(H_1,H_2)$. The EMD distance is known to be, in general, a robust way to compare two histograms [RTG00]. In our case, d_{EMD} creates fewer false negatives than d_H, but still too many of them. A similar effect has been reported for CBSR by [SST*15].

Median value separation: Studying the above-mentioned false negatives, we noted that they are mainly caused by small differences in the location (cut-length) of the largest peaks (longest bars H^i) in the two compared histograms. These correspond to slightly different thickness-distributions of the two compared shapes Ω_1 and Ω_2. To alleviate this, we work as follows: For a shape Ω, we compute its median cut-length value m, and use it to divide its histogram H into two sub-histograms H^l and H^r which contain cuts shorter or equal than, respectively larger than, m. For H^l, we list bars in decreasing cut-thickness order; for H^r, bars are listed in increasing thickness order. Both histograms H^l and H^r use, together, the same fixed number of bins B as the original H. When comparing two shapes Ω_1 and Ω_2, we pad H_1^l and H_1^r with zero-size bins to the right so as to match the corresponding sizes of their counterparts H_2^l and H_2^r. Note that right-padding is made possible by the reordering of bars in H^l mentioned above. Finally, we use the EMD metric to compare the corresponding histogram-pairs, yielding the distance metric $d_{MV}(H_1,H_2) = \frac{1}{2}[d_{EMD}(H_1^l,H_2^l) + d_{EMD}(H_2^r,H_2^r)]$. Figure 7 middle shows this for the shapes, for which we get $d_{MV} = 0.281$, which reflects much better their visual similarity than d_{EMD} and d_H studied before.

Multilevel distance: As outlined above, small changes in the locations of tall bars in cut-space histograms can create large undesired distance differences. The median-value distance d_{MV} partially fixes this by effectively 'aligning' two histograms along their median values. We next refine this fix to also consider *gaps* in a histogram. These are near-empty bins separating blocks of non-empty bars (see Fig. 5), and correspond to cut sizes appearing rarely in a

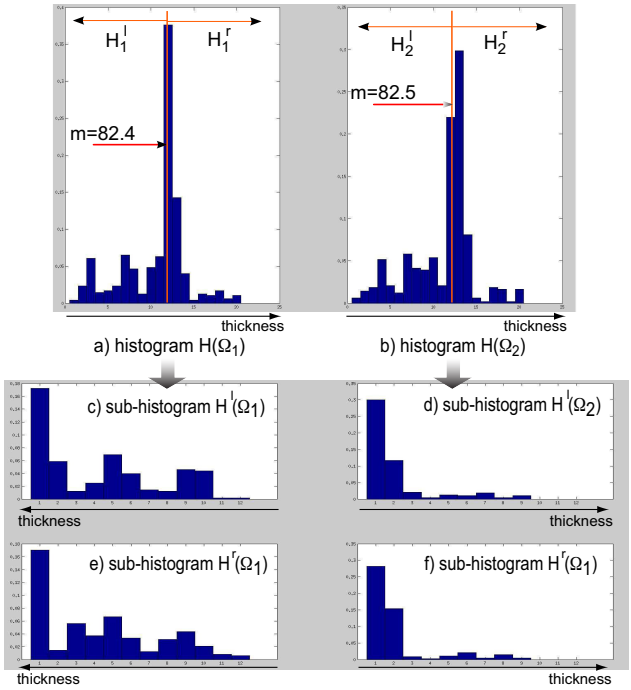

Figure 4: *Median value separation histogram-distance d_{MV}.*

shape. Since a gap is, by definition, surrounded by two blocks of non-empty bars, it corresponds to shape zones showing rapid local-thickness transitions, such as joints between parts. Gaps were used in [FJT15] to segment shapes along these joints. In contrast, we use gaps to compare shapes more robustly, as follows. First, we define gaps g^i in a histogram H as bars H^i containing less than $\tau\|CS\|$ of the total number of cuts, where $\tau = 0.01$ has given good results for all our experiments. Gaps effectively partition H into several bin-blocks B^i, each being defined by two consecutive gaps (g^i_l, g^i_r) to its left, respectively right. Given two such histograms H_1 and H_2, we compare their block-sets $B_1 = \{B^i_1\}$ and $B_2 = \{B^i_2\}$ using d_{MV} on block-pairs in B_1 and B_2 whose center distance $|g^i_l + g^i_r|/2$ is less than $\beta = 3B/20$ bars, which corresponds to three bars for our preset $B = 20$ number of bars in a histogram. Note that the complexity of this comparison is $O(\beta \cdot \max(|B_1|, |B_2|))$, *i.e.*, basically linear in the number of blocks, since a block in, say, B_1 only gets compared worst-case with 2β other blocks in B_2. Denoting the set of such matching blocks by $BP \subset B_1 \times B_2$, we get our multilevel distance metric $d_{ML}(H_1, H_2) = \sum_{(B_i,B_j)\in BP} d_{MV}(B_i, B_j)$. Figure 5 shows this idea for our two plier shapes (Fig. 3), whose histograms show one gap each, and thus have been cut into two blocks. Intuitively, we see that block-detection 'aligns' two histograms independently from the longest-bar alignment provided by d_{MV}. For our plier shapes, we obtain $d_{ML} = 0.0321$, which reflects the (very high) visual similarity of the two shapes better than all metrics considered so far.

4. Implementation and results

We implemented our shape descriptor using the 3D skeletonization algorithm in [JST15], which is one of the fastest, most robust, exact, and easy to use methods to extract regularized 3D-multiscale surface-skeletons in existence. Cut spaces CS were built following

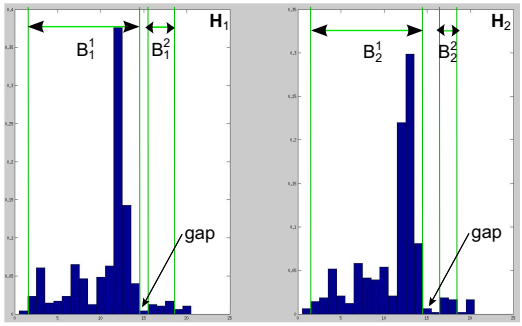

Figure 5: *Multilevel histogram-distance d_{ML}.*

the method in [FJT15]. To test our proposed descriptor, we chose to use the McGill 3D Shape Benchmark [CIM15], which is well-known in the shape retrieval and computer vision community. The database contains 309 shapes, grouped into 13 semantic classes. Each class contains objects having various scales, poses, and articulations (see Fig. 6 for a sample subset). Since both [JST15] and [FJT15] are voxel-based methods, we voxelized all our test shapes to resolutions ranging between 200^3 and 500^3 voxels, using *binvox* [NT03]. Our entire pipeline, written in C++, was executed on 3.5 GHz Linux PC.

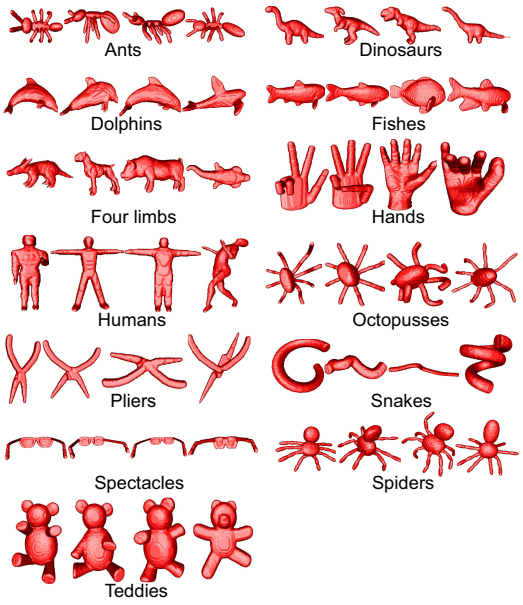

Figure 6: *Example shapes from the used benchmark [CIM15].*

Table 1 shows the performance of our cut-space-based descriptor using the three distance metrics in Sec. 3.4. To test retrieval accuracy for the used database, we used two methods:

Fine grained: Given a shape $\Omega \in D$ from a database D, we compute the query's *precision* for a query-size $Q = 10$, *i.e.*, find the most similar 10 other shapes $\Omega_i \in D$ according to $d(H(\Omega), H(\Omega_i))$, and count how many of these are in the same class as D.

Coarse grained: We apply the same procedure as above, but merge

Classes	Class sizes	d_{EMD} (fine)	d_{EMD} (coarse)	d_{MV} (fine)	d_{MV} (coarse)	d_{ML} (fine)	d_{ML} (coarse)
humans	29	0.807	0.945	0.755	0.831	0.776	0.941
hands	20	0.345	0.7	0.36	0.6	0.435	0.81
four limbs	31	0.494	0.813	0.439	0.771	0.442	0.858
dinosaurs	18	0.439	0.844	0.389	0.828	0.339	0.878
spiders	31	0.787	0.900	0.777	0.974	0.797	0.939
ants	30	0.793	0.837	0.62	0.877	0.737	0.803
octopus	25	0.608	0.852	0.652	0.828	0.66	0.956
fishes	23	0.517	0.709	0.522	0.704	0.674	0.791
dolphins	12	0.4	0.683	0.467	0.642	0.442	0.642
pliers	20	0.92	0.92	1	1	0.87	0.87
teddy	20	0.72	0.72	0.775	0.775	0.885	0.885
spectacles	25	0.292	0.292	0.34	0.34	0.432	0.432
snakes	25	1	1	0.684	0.684	0.624	0.624

Table 1: *Shape retrieval results with three for the distance metrics d_{MV}, d_{EMD}, and d_{ML}.*

classes having structurally similar objects into three so-called superclasses: limbed objects (humans, hands, dinosaurs, four-limbs, teddies); highly articulated objects (spiders, ants, octopuses, spectacles); and low-articulated objects (fishes, dolphins).

We applied both above tests to all 309 shapes in the database and then averaged the results per shape-class. Table1 shows the results. Figure 7 gives more detailed insights, showing the six most-similar shapes retrieved for a query shape for the distance metrics d_{ML}, d_{EMV}, and d_{MV}. The Hellinger distance d_H was left out as it yielded significantly poorer results. From these data, we see that d_{ML} gives the visually best results, followed by d_{MV} and next by d_{EMD}.

Noise robustness: To test the robustness of our descriptor to noise, we created a database where noise was added to the shapes in [CIM15], in the form of Gaussian bumps of two heights (small – 3% of the shape diameter $\Phi(\Omega)$, and large – 6% of $\Phi(\Omega)$); and two standard deviations (small – 4% of $\Phi(\Omega)$, and large – 8% of $\Phi(\Omega)$). Noise bumps were Poisson distributed over $\partial\Omega$. We next queried for a clean shape, and observed that we got both noised versions thereof and clean versions of shapes in the same class as the top-hits (Fig. 8 top row). We also queried the four types of noised shapes corresponding to the clean shape Ω, and noticed that we got, in three cases, three of the noised variants of Ω, and in the fourth case all four variants (Fig. 8, bottom 4 rows). In all cases, the clean shape appeared in the top-six most similar retrieved shapes. This shows that our descriptor is both robust to noise (it retrieves the correct same-class shapes and does not introduce false-positives because of noise); and also sensitive to shape (it retrieves most of the noised versions of the input shape).

5. Discussion

We nest discuss several relevant aspects of our proposed descriptor.

Robustness: The proposed descriptor is rotation, translation, scaling, pose, noise, and voxelization-resolution invariant. This is guaranteed by the corresponding properties of the underlying skeleton cut-space and histogram normalization (Sec. 3.3). Note that, while these properties hold by construction in the continuous space $\mathbb{R}^3$, they are preserved in the voxel space $\mathbb{Z}^3$ only as well as the underlying skeletonization method can manage to do so. The method we use here achieves very good results in this respect, as discussed in detail in [JST15]. Note also that, to ensure similar

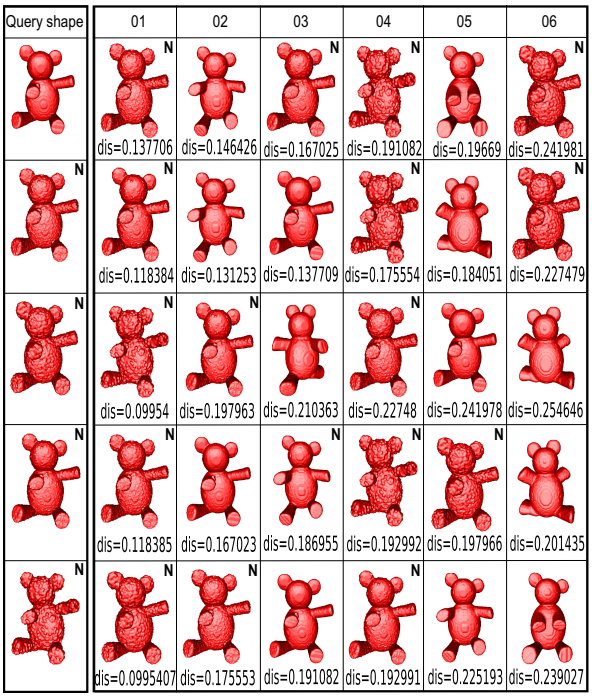

Figure 8: *Retrieval robustness in presence of noise. Shapes marked 'N' contain artificially-added noise.*

properties, other CBSR methods require more complex techniques, such as spherical harmonics [LKyS03, SV01, KFR03, KFR04] or delicate, and time-consuming, manual model alignment [TKA12].

Comparison: Our evaluation, while limited, is in line with several related papers. Osada [OFCD02] uses a database of 133 models grouped in 25 classes. For evaluating the performance of their method, they measure the query precision, and the percentage of all queries where the top match was from the queried shape's class, and obtain values of 30%, respectively 70%. In comparison, our corresponding values are of 70% (for the d_{ML} metric), respectively 100% (all metrics) (see Tab. 1). Furthermore, the spectral descriptors of Jain and Zhang [JZ07] yield 70...75% precision performance and outperform both the light-field [CTSO03] and spherical-harmonic [KFR03], whose precision is 68% and 60%, respectively. On the same dataset, other query-precision results include 49% – shape-distributions [OFCD02], 45% – covariance [TL15] and 50% – diffusion [FXD*15], as reported in [FXD*15, LGS10]. When comparing the above figures with ours, we should stress that the obtained insights are, of course, limited by the fact that we use different databases. This is due to the lack of access to the above-mentioned implementations (so we couldn't use them on our database) and the difficulty and cost to convert the databases used in the above papers, when these were publicly accessible, to our voxel representation (so we couldn't easily use our implementation on these databases).

The skeleton-based descriptor in [SKK04] was tested on two databases (99 shapes, 9 classes; and 144 shapes, 8 classes). Its accuracy, measured by the so-called first tier (precision with the query size equal to the class size of the queried shape), yielding a value of 97% on average – in our case, the corresponding average value is

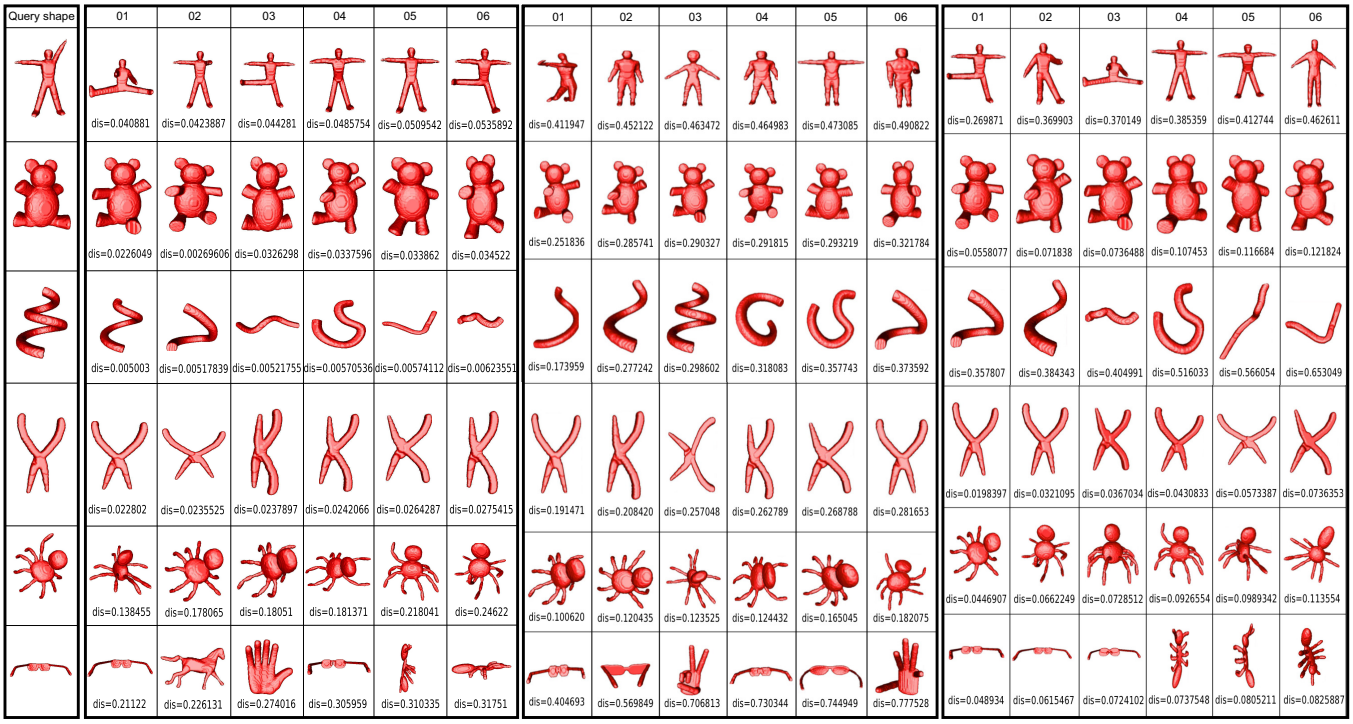

Figure 7: *Examples of shape retrieval results for the distance metrics d_{EMD} (left), d_{MV} (middle), and d_{ML} (right). For each distance metric, shapes are listed from left to right in increasing distance order. Actual distance values are indicated under the retrieved shapes.*

of 80%. However, we note that [SKK04] can only treat 2D shapes, which are known to be much easier to match than 3D shapes. Similarly, the skeleton-based path similarity matching method in [BL08] shows quite good retrieval results, with an average precision of 95% for a query size of 10 shapes, but is also limited to 2D shapes. Sundar [SSGD03] tested their 3D curve-skeleton-based descriptor on a database of 100 shapes. However, while a few retrieval results are shown, no statistics are provided. Cornea *et al.* compared their 3D curve-skeleton-based descriptor on a 1000 shape database [CDS*05]. They report a first-tier value of 17% and nearest-neighbor value of 71%. While both [CDS*05] and our method use the EMD to compare shape descriptors, [CDS*05] uses the entire curve-skeleton as descriptor, while we use the cut-space thickness-histogram. The latter is significantly more invariant to irrelevant shape changes than the former, which explains, at least partially, our better results.

Efficiency and ease of use: The proposed descriptor can be computed fully automatically, with no user parameters needed to be set or tuned. For a shape Ω, its computation complexity is $O(\|S_{\partial\Omega}\|D)$, where $D = \max_{\mathbf{x}\in\Omega} DT_{\partial\Omega}(\mathbf{x})$ indicates the maximal shape thickness and $\|S_{\partial\Omega}\|$ the number of voxels of the shape's surface skeleton. In practice, this allows computing our descriptor in subsecond times on shapes up to a few hundred voxels cubed on the platform indicated in Sec. 4. Significant speed-ups can be easily achieved, if desired, by trivially parallelizing the cut space computation on the CPU or GPU, if desired.

Limitations: As outlined earlier, our descriptor essentially captures the local shape thickness only. As such, its discriminative power is lower, in general, than more advanced descriptors, in general. However, for a fair comparison, one should relate our proposal chiefly to other thickness-based descriptors in CBSR (or, more generally, descriptors with the same small size, in our case, $B = 20$ values). Indeed, comparing descriptors of highly different sizes can be perceived as unfair, since these have highly different information content. Separately, our proposal can be directly used to replace other less accurate thickness descriptors known in the literature, *e.g.* [SST*15, LKyS03]. A second limitation relates to the benchmark we used [CIM15], which contains mainly locally-tubular shapes. For a better insight, more varied shapes should be added to such a benchmark, a task which we consider for future work. Finally, besides precision values, CBSR benchmarks also typically use nearest-neighbor and first-tier values [CDS*05, SKK04]. We plan to add these metrics in our future evaluations of our proposed descriptor.

6. Conclusions

We have presented a new method for characterizing 3D shapes via a thickness-histogram descriptor. Our descriptor captures the *local* thickness of a 3D shape in an accurate and computationally-savvy way, by using the 3D surface-skeleton of a shape to construct a regularized space of smooth cuts which are tight around the shape, and also optimally oriented across the local shape curve-skeleton to capture the shape's local thickness. We next studied four histogram-distance metrics to compare such descriptors, and showed that two such novel metrics achieve shape matches which reflect the visual

similarity of 3D shapes in better ways than classical histogram-comparison metrics used in the literature. Our method is simple to implement, fully automatic, and robust to shape variations such as pose, scale, rotation, noise, and voxelization resolution.

Future work directions consider the integration of additional shape information in our cut histograms, such as the cut shape, eccentricity, and orientation, as well as more detailed testing our descriptor on larger shape benchmarks, and its (easy) integration with other existing shape descriptors in the literature.

References

[BBK*10] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R., MAHMOUDI M., SAPIRO G.: A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *IJCV 89*, 2-3 (2010), 266–286. 2

[BL08] BAI X., LATECKI J.: Path similarity skeleton graph matching. *IEEE TPAMI 30*, 7 (2008). 2, 7

[BLL07] BAI X., LATECKI J., LIU W.-Y.: Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE TPAMI 29*, 3 (2007), 449–462. 2

[Blu67] BLUM H.: *A transformation for extracting new descriptors of shape*. Models for the perception of speech and visual form. MIT Press, 1967. 2

[CDS*05] CORNEA N., DEMIRCI M., SILVER D., SHOKOUFANDEH A., DICKINSON S., KANTOR P.: 3D object retrieval using many-to-many matching of curve skeletons. In *Proc. IEEE SMI* (2005), pp. 147–152. 1, 2, 7

[CIM15] CIM MCGILL: McGill shape benchmark, 2015. www.cim.mcgill.ca/~shape/benchMark. 5, 7

[CSM07] CORNEA N. D., SILVER D., MIN P.: Curve-skeleton properties, applications, and algorithms. *IEEE TVCG 13*, 3 (2007), 87–95. 1, 2

[CTSO03] CHEN D.-Y., TIAN X.-P., SHEN Y.-T., OUHYOUNG M.: On visual similarity based 3D model retrieval. *CGF 22*, 3 (2003), 223–232. 2, 6

[DS06] DEY T., SUN J.: Defining and computing curve-skeletons with the medial geodesic function. In *Proc. SGP* (2006), pp. 143–152. 2, 3

[FJT15] FENG C., JALBA A., TELEA A.: Part-based segmentation by skeleton cut space analysis. In *Proc. ISMM* (2015), pp. 324–336. 3, 5

[FLM03] FOSKEY M., LIN M., MANOCHA D.: Efficient computation of a simplified medial axis. *Proc. ACM Symp. Solid Modeling* (2003), 96–107. 3

[FMKC03] FUNKHOUSER T., MIN O., KAZHDAN M., CHEN J.: A search engine for 3D models. *ACM TOG 2*, 1 (2003), 83–105. 1

[FXD*15] FANG Y., XIE J., DAI G., WANG M., ZHU F., XU T., WONG E.: 3D deep shape descriptor. In *Proc. IEEE CVPR* (2015), pp. 2319–2328. 2, 6

[GF15] GETTO R., FELLNER D.: 3D object retrieval with parametric templates. In *Proc. 3DOR* (2015). 1

[GK04] GIBLIN P., KIMIA B.: A formal classification of 3D medial axis points and their local geometry. *IEEE TPAMI 26*, 2 (2004), 238–251. 3

[JKT13] JALBA A. C., KUSTRA J., TELEA A.: Surface and curve skeletonization of large 3D models on the GPU. *IEEE TPAMI 35*, 6 (2013), 1495–1508. 1, 2, 3

[JST15] JALBA A., SOBIECKI A., TELEA A.: An unified multiscalse framework for planar, surface, and curve skeletonization. *IEEE TPAMI 38*, 1 (2015), 30–45. 2, 3, 5, 6

[JTV08] JOHAN W., TANGELDER H., VELTKAMP R.: A survey of content based 3D shape retrieval methods. *Multimed Tools Appl 39*, 3 (2008), 441–471. 1, 4

[JZ07] JAIN V., ZHANG H.: A spectral approach to shape-based retrieval of articulated 3D models. *Comput. Aided Des. 39*, 5 (2007), 398–407. 2, 6

[KFR03] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proc. SGP* (2003), Eurographics, pp. 156–164. 1, 2, 6

[KFR04] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Symmetry descriptors and 3D shape matching. In *Proc. SGP* (2004), pp. 46–54. 2, 6

[LGS10] LIAN Z., GODIL A., SUN X.: Visual similarity based 3D shape retrieval using Bag-of-Features. In *Proc. SMI* (2010), pp. 25–36. 2, 6

[LKyS03] LIU X., KANG S. B., YEUNG SHUM H.: Directional histogram model for three-dimensional shape similarity. In *Proc. IEEE CVPR* (2003), pp. 813–820. 2, 3, 4, 6, 7

[NT03] NOORUDDIN F., TURK G.: Simplification and repair of polygonal models using volumetric techniques. *IEEE TVCG 9*, 2 (2003), 191–205. 5

[OFCD02] OSADA R., FUNKHOUSER T., CHAZELLE B., DOBKIN D.: Shape distributions. *ACM TOG 21*, 4 (2002), 807–832. 2, 6

[RH08] ROERDINK J., HESSELINK W.: Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform. *IEEE TPAMI 30*, 12 (2008), 2204–2217. 3

[RTG98] RUBNER Y., TOMASI C., GUIBAS L. J.: A metric for distributions with applications to image databases. In *Proc. IEEE ICCV* (1998), pp. 59–66. 2, 4

[RTG00] RUBNER Y., TOMASI C., GUIBAS L. J.: The earth mover's distance as a metric for image retrieval. *IJCV 40*, 2 (2000), 99–121. 4

[RvWT08a] RENIERS D., VAN WIJK J. J., TELEA A.: Computing multiscale skeletons of genus 0 objects using a global importance measure. *IEEE TVCG 14*, 2 (2008), 355–368. 2

[RvWT08b] RENIERS D., VAN WIJK J. J., TELEA A.: Computing multiscale skeletons of genus 0 objects using a global importance measure. *IEEE TVCG 14*, 2 (2008), 355–368. 3

[SHP97] S.BERCHTOLD, H.-P.KRIEGEL: S3: Similarity search in CAD database systems. In *In ACM SIGMOD* (1997), pp. 564–567. 1

[SJT14] SOBIECKI A., JALBA A., TELEA A.: Comparison of curve and surface skeletonization methods for voxel shapes. *Pattern Recogn Lett 47* (2014), 147–156. 2

[SK96] SIDDIQ K., KIMIA B.: A shock grammar for recognition. In *Proc. IEEE CVPR* (1996), pp. 507–513. 2

[SKK04] SEBASTIAN T., KLEIN P., KIMIA B.: Recognition of shapes by editing shock graphs. *IEEE TPAMI 26*, 5 (2004), 550–571. 1, 2, 6, 7

[SOG09] SUN J., OVSJANIKOV M., GUIBAS L.: A concise and provably informative multi-scale signature based on heat diffusion. *CGF 29*, 5 (2009), 1383–1392. 2

[SSDZ99] SIDDIQI K., SHOKOUFANDEH A., DICKINSON S., ZUCKER S. W.: Shock graphs and shape matching. *IJCV 35*, 1 (1999), 13–32. 2

[SSGD03] SUNDAR H., SILVER D., GAGVANI N., DICKINSON S.: Skeleton based shape matching and retrieval. In *Proc. ACM SMI* (2003), pp. 130–137. 1, 2, 7

[SST*15] SCHMIDT W., SOTOMAYOR J., TELEA A., SILVA C., COMBA J.: A 3D shape descriptor based on depth complexity and thickness histograms. In *Proc. SIBGRAPI* (2015), pp. 433–440. 2, 3, 4, 7

[SV01] SAUPE D., VRANIC D.: 3D model retrieval with spherical harmonics and moments. In *Proc. DAGM* (2001), Springer, pp. 392–397. 2, 4, 6

[TKA12] TATSUMA A., KOYANAGI H., AONO M.: A large-scale shape benchmark for 3D object retrieval: Toyohasi shape benchmark. In *Proc. APISPA ASC* (2012), pp. 1–10. 2, 6

[TL15] TABIA H., LAGA H.: Covariance-based descriptors for efficient 3D shape matching, retrieval, and classification. *IEEE Transactions on Multimedia 17*, 9 (2015), 1591–1603. 2, 6

[XHS08] XIE J., HENG P., SHAH M.: Shape matching and modeling using skeletal context. *Patt Recog 41* (2008), 1756–1767. 1, 2