

Sphere intersection 3D shape descriptor (SID) [☆]



Kirill Pevzner ^{*}, Andrei Sharf, Jihad El-Sana

Ben Gurion University, Israel

ARTICLE INFO

Article history:

Received 2 April 2015

Received in revised form 31 August 2015

Accepted 1 September 2015

Available online 15 September 2015

Keywords:

Mesh processing

Descriptor

Signature

Similarity

Retrieval

ABSTRACT

This paper presents a novel 3D shape descriptor which explicitly captures the local geometry and encodes it using an efficient representation. Our method consists of multiple evolving fronts which are realized by a set of growing spheres on the surface. At the core of this method is a simple intersection operator between the spheres and the shape's surface. Intersection curves yield a discrete sampling of the surface at different positions and scales. Our key idea is to define a shape descriptor that captures the continuous local geometry of the surface in an efficient and consistent representation by intersecting the surface with multiple spheres and transforming the intersection curve to frequency domain. To evaluate our descriptor, we define shape similarity metric and perform shape matching on the SHREC11 non-rigid benchmark and other classes.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Databases of 3D models are becoming nowadays available in a number of disciplines, including computer graphics, CAD, and medicine (Veltkamp and Hagedoorn, 2001). As a result of the extensive growth of 3D databases, shape-retrieval techniques are becoming valuable tools for analysis and discovery. The SHREC contest and benchmark (Boyer et al., 2011) have been stimulating forces in this field.

A common method for matching objects is to represent each object as a set of local shape features sampled from the object's surface. For a pair of objects and a given correspondence between their features, similarity is measured by comparing the corresponding features using a cost function (e.g., Belongie et al., 2001; Chua and Jarvis, 1997; Johnson and Hebert, 1999). Nevertheless, shapes and features can be represented in many different ways, and finding an appropriate representation that is amenable to surface matching is still an open research problem.

Our descriptor is inspired by the Shape Context descriptor recently explored by Kokkinos et al. (2012). Similarly, we sample the continuous surface into a discrete set of samples. Nevertheless, instead of discretization into bins, we utilize continuous 1D surface curves as a more informative representation of the shape. Our idea is that shape characteristics, especially subtle ones require high-order descriptions to represent them appropriately. For example, the tip of the octaflower in Fig. 2 (right) is a complex feature and singular point which cannot be described using principal curvatures. Using our descriptor, we represent the region using explicit 1D curves sampling the surface, thus capturing the feature as can be seen in the graph.

At the core of our shape descriptor computation is a simple sphere and surface intersection operation. Essentially, we grow spheres on the surface and compute their intersection with the surface at several predefined scales. The intersections

[☆] This paper has been recommended for acceptance by Pierre Alliez.

^{*} Corresponding author. Tel.: +972558836169.

E-mail address: kirillpe@gmail.com (K. Pevzner).

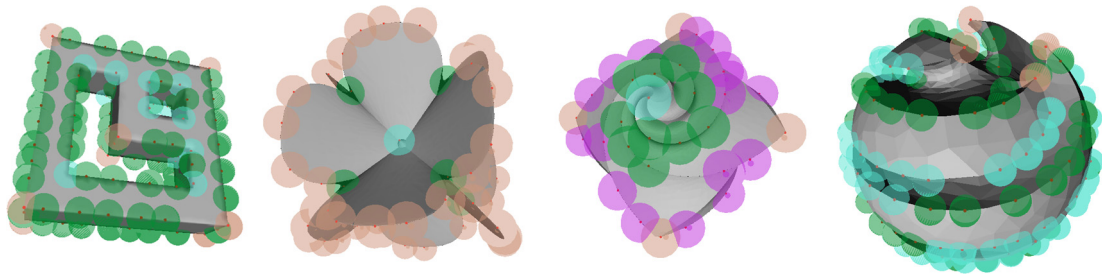


Fig. 1. Features detected using our *SID* shape descriptor for several models of different classes. Sphere colors denote self similarity in each model. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

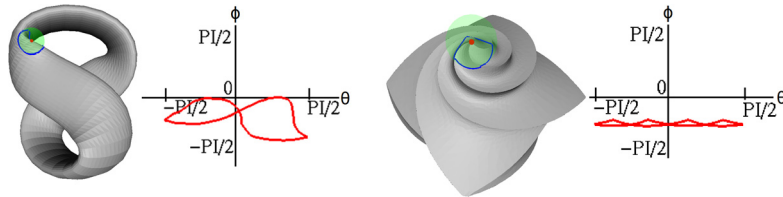


Fig. 2. Two shapes with complex features (knot-left, octaflower-right), such as curvature extrema, saddle and singular points. For each green sphere on shapes, graphs to the right show the parameterized intersection curve. Note the large differences in the graphs between the saddle point (left) and singular tip point (right). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

result in a set of parameterized curves that represent excerpts of the local surface. For each point we accumulate the set of curves. The set of curves is then transformed to the frequency domain, resulting in a compact yet effective 2D representation of local 3D geometry.

We define a shape similarity metric based on the correlation between “interesting shape features” and curvature, following the observation of Marr (1982). To compare two shapes, we filter out smooth or flat regions, and compare the curved regions by measuring the distance between their descriptors.

We evaluate our work by performing both qualitative and quantitative analysis. For qualitative evaluation we use multiple shapes with different characteristics and deformations (see Figs. 2, 8 and 1). For quantitative evaluation we use the SHREC 11 non-rigid benchmark (Boyer et al., 2011) and SHREC 12 abstract shapes benchmark (Biasotti et al., 2012). The results show that *SID* outperforms most state-of-the-art algorithms used in the comparison in terms of retrieval rates. With regards to MeshSIFT (Smeets et al., 2013), although retrieval rates are similar to us, we observe that *SID* holds favorable properties in terms of speed, meaningful keypoint selection and consistency of keypoint selection.

Our contributions are as follows:

- A descriptor which captures the shape via explicit sampling of the surface with continuous 1D curves at keypoints. Our descriptor is informative and distinctive (see Fig. 2).
- An efficient and meaningful feature detection scheme based on our descriptor formulation.
- A shape retrieval scheme which offers a unique balance and combination of correctness, speed and handling multiple classes of shapes.

2. Related work

Shape descriptors and similarity have been extensively studied in fields such as computer-vision, CAD and molecular-biology (Veltkamp and Hagedoorn, 2001). In 2D images and video, shape matching has been broadly researched (Flickner et al., 1995; Jacobs et al., 1995; Ogle and Stonebraker, 1995; Castelli and Bergman, 2001) and is beyond the scope of this work. A major problem in extending these systems to 3D is that 3D models typically lack an underlying regular parametric domain. For example, 2D shapes have a natural arc length parametrization of their boundary contours while 3D surfaces of arbitrary genus do not. As a result, common shape descriptors for 2D contours (e.g., Arbter et al., 1990; Arkin et al., 1990; Kashyap and Chellappa, 1981) cannot be directly extended to 3D surfaces.

Nevertheless, descriptors for 3D shape matching have been studied for more than two decades. Gaussian images (Horn, 1984) and spherical representations (Delingette et al., 1993) are global representations useful for describing simple objects. Stein and Medioni (1992) recognize 3D objects by matching points using structural indexing and their “splash” representation. Similarly, Chua and Jarvis match points to align surfaces using principal curvatures (Chua and Jarvis, 1996) and “point-signatures” (Chua and Jarvis, 1997). The Harmonic shape images (Zhang and Hebert, 1999) map a patch that is topo-

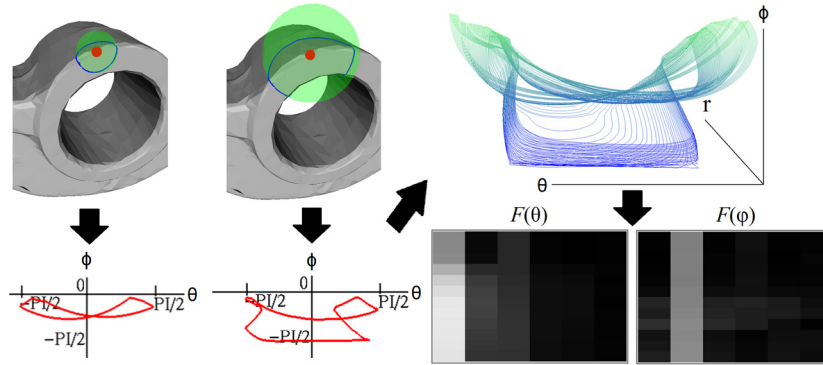


Fig. 3. Sphere Intersection Descriptor (SID) of a single feature overview. Upper left: two sphere intersections with different radii. Bottom left: corresponding parameterized intersection curves. Upper right: multiple intersections aggregated. Bottom right: frequency domain transformation yielding a compact 2D image representation.

logically equivalent to a disk into a 2D domain. Similar to this work, Spin Images (Johnson and Hebert, 1999) encode the local patch geometry in a compact 2D image using cylindrical coordinates.

Our descriptor significantly differs from these methods as it encodes continuous surface excerpts around a point. Thus, instead of analyzing local properties (for example curvature), we capture the actual surface by encoding curves around a point. This representation benefits from both being compact and yet descriptive. Nonetheless, our formulation is simple, requiring only to compute a surface intersection with a sphere and is not restricted to disk-like patch topology.

Another approach for describing shapes is indexing their features based on statistical properties. Histograms of geometric statistics have been considered (Ankerst et al., 1999; Osada et al., 2001; Ohbuchi et al., 2003) as well as harmonic-based representations (Funkhouser et al., 2003; Kazhdan et al., 2003). Ankerst et al. (1999) propose global histograms encoding shells and sectors around a model's centroid. Osada et al. (2001) represent shapes with probability distributions of geometric properties computed from random points. Kazhdan et al. (2003) represent the global shape using spherical harmonics, for rotational invariant descriptors. Giorgi et al. (2010) utilize best view selection to generate descriptive 2D binary images from 3D models for effective matching. While these methods propose alternate surface representations, ours encodes the explicit surface. Hence, it retains all surface properties and characteristics.

Multi-scale features similar to 2D SIFT (Lowe, 2004) have been explored in the context of 3D shapes (Li and Guskov, 2005; Zaharescu et al., 2009; Smeets et al., 2013). These techniques develop local shape descriptors based on differential properties and scalar functions of the mesh surface. MeshSIFT (Smeets et al., 2013) detects features by smoothing the mesh at various degrees and seeking for local minima/maxima of mean curvature. The feature description is performed by constructing a histogram of slant and angle per each scale in scale space.

Zaharescu et al. (2009) present MeshHOG which seeks Laplacian extrema in a scalar function defined on a surface. Sun et al. (2009) and Bronstein and Kokkinos (2010) utilize heat-kernels to compute the bending invariant shape signatures. The use of heat-kernels has many benefits such as mathematically provable properties and detection of small number of meaningful features but is also time consuming and implicit representations of the surface. Pottmann et al. (2009) analyze the stability of differential properties on surfaces with regard to their scale. Sipiran and Bustos (2011) extend the 2D Harris operator to 3D meshes, they analyze the k-ring of vertices around a point in order to form the descriptor and propose two modes of operation for feature detection: choosing points with highest Harris value or clustering. The first method suffers from choosing multiple features on sharp edges of the surface thus being somewhat ambiguous in the feature selection result. The second method suffers from lesser repeatability according to the authors. Darom and Keller (2012) show scale invariant extensions of the SIFT and Spin Images descriptors to 3D meshes. Similar to these works, we also define our shape descriptor in a scale-space context. Nevertheless, our signature is an explicit representation of the surface and not an implicit measurement of a scalar function, shape distribution or differential properties. Thus, our representation is less sensitive to noise and surface inconsistencies.

In a seminal work, Tombari et al. (2013) evaluated and compared 3D feature descriptors. A major conclusion was that the number of keypoints do not correlate with matching success. For example, KPQ (Mian et al., 2010) finds many keypoints and yields good matching scores in contrast to HKS (Sun et al., 2009) and Salient Points (Castellani et al., 2008) which both find very few keypoints and both sub-performed. We come to the conclusion that detecting a small number of meaningful features that produce good retrieval results is still a major challenge. Our method specifically addresses this challenge.

Mortara et al. (2004) explore different characteristics of intersection curve resulting from intersecting a sphere with a triangle mesh. Some of the characteristics considered are: concavity vs. convexity, curve length and number of connected components. Their research focused on segmentation of shapes rather than shape retrieval, and did not address the question of fine grain dissimilarity between features. We choose to further explore the sphere to mesh intersection curve with respect to mesh retrieval application.

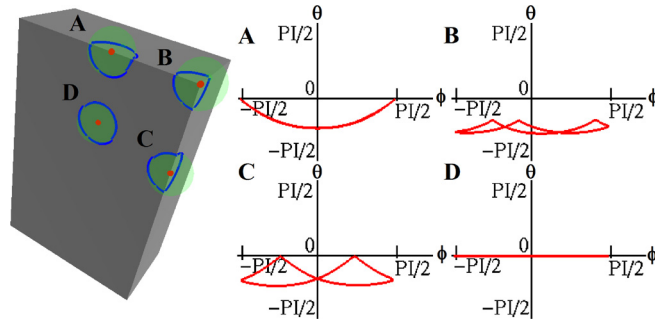


Fig. 4. (Left) A simple box with 4 spheres (green) and intersection curves (blue). (Right) Intersection curve projection to spherical coordinates as $(\phi(t), \theta(t))$ (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3. Overview

Our framework consists of the following main components: feature detection, feature description, shape similarity. In the following we provide an overview of each component.

- **Feature detection:** the selection of features is done iteratively with spheres of growing radius, centered at mesh vertices. At each step of the feature detection loop we compute a saliency metric on the sphere intersection curve with the surface. Intersecting spheres are detected and only spheres with stronger metric are kept for the next iteration. Fig. 6 demonstrates the iterative detection process.
- **Feature description:** given a keypoint and a support sphere radius from the previous step, the sphere intersection curve with the surface is projected to spherical coordinates and parameterized to $\phi(t), \theta(t)$ components. Lastly, $\phi(t), \theta(t)$ are transformed to frequency domain thus producing rotational invariant local feature descriptor. Fig. 3 demonstrates the sphere intersection with the mesh, the resulting curve on the (ϕ, θ) plane and the transformation to frequency domain.
- **Shape similarity:** for similarity computation between two meshes, we perform a min cost maximal matching on a bipartite graph generated from the set of descriptors of each mesh. Similarity between two given feature descriptors is given by L_1 distance of their magnitudes. Thus, model correspondence is performed by minimizing the cost of the matching on the feature descriptors sets.

4. Technical details

In the following sections we discuss each component in detail.

4.1. Feature detection

Feature point detection is done in an iterative manner. We start with all mesh vertices as candidates for being considered feature points and reduce the amount of candidates in each iteration. In each iteration of the feature detection loop we perform the following for each candidate vertex v :

1. Extract the Sphere-with-Surface intersection curve.
2. Transform the intersection curve to spherical coordinations.
3. Parameterize the intersection curve as $(\phi(t), \theta(t))$.
4. Compute of saliency metric.

In the end of each iteration we consider the intersection of neighboring spheres and upon sphere-to-sphere intersection we keep only the one with the stronger saliency metric. Thus, the algorithm converges from all mesh vertices to keypoints.

4.1.1. Sphere intersection curve

Given a vertex v and radius r we compute the intersection of a sphere of radius r centered at v with the mesh surface. This gives a closed loop of adjacent faces that intersect with the sphere. By choosing the initial radius to be small enough, we enforce only one closed loop per sphere in the first iteration. Then, each iteration we increase r and compute a new sphere intersection curve. In case that the sphere intersection produces several closed loops of faces we take the faces loop closest to the loop from the previous iteration. it follows by induction that we always have one closed loop and that it will not jitter between different components.

The faces are sampled uniformly so we obtain the intersection curve points in global Euclidean coordinate system (see Fig. 5). Then we impose a local Euclidean coordinate system UVW , with origin at v and its W axis as the *consistent normal* at v . We arbitrarily set the UV axes to be orthonormal to W .

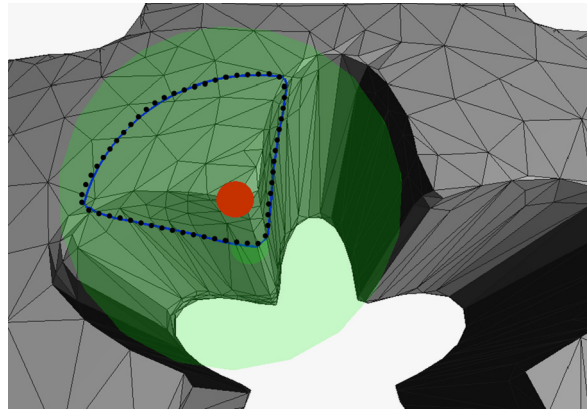


Fig. 5. Sphere intersection illustration showing the triangle intersection with arc and its sampling.

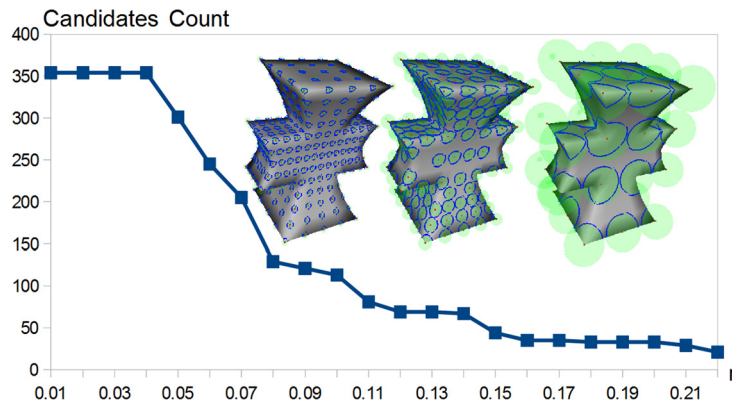


Fig. 6. A graph showing the number of spheres as a function of sphere radii. This essentially demonstrates the progress of our iterative algorithm for feature detection. Three snapshots of the state of the feature detection are also provided, at radii 3%, 7%, 20% of mesh size.

The consistent normal of v is computed as follows:

$$cn(v) = \begin{cases} normal(v), & \text{if } \|v - icc\| < t \\ sign((v - icc) \cdot normal(v)) \cdot \frac{v - icc}{\|v - icc\|}, & \text{else} \end{cases} \quad (1)$$

where icc is the intersection curve centroid and t is a threshold. The first case sets the consistent normal of v to be the normal of v if the intersection curve centroid is too close to v (determined by threshold t). The purpose of this case is to prevent “jitter” of the consistent normal at small scales. The second case sets the consistent normal of v to be the vector connecting the intersection curve centroid and v , we enforce the consistent normal to be in the same direction of the original normal of v by multiplying with $sign((v - icc) \cdot normal(v))$. The construction of the consistent normal follows the direction of the original normal and gives a normal that is invariant to different deformations: non-rigid, local noise and various triangulations. Thus, we obtain a stable local coordinate system for locally similar surfaces.

Next follows the transformation of c points to local spherical coordinate system. Since r is constant, we can represent c as two 1D functions defined by (θ, ϕ) , where $\phi, \theta \in [-\pi/2, \dots, \pi/2]$. We parameterize the intersection curve by performing a walk on the curve c and encoding the angles (θ, ϕ) with respect to the axes. The intersection curve points are transformed into spherical coordinates following the formula:

$$\phi = \sin(\text{atan2}(p_x, p_y)) \quad (2)$$

$$\theta = \text{asin}(p_z) \quad (3)$$

where (p_x, p_y, p_z) is a point on c in local Euclidean coordinate system. We apply a \sin function to ϕ as we require a cyclic function for the Fourier transform. The origin of ϕ (i.e., $\phi = 0$) is determined by the rotational orientation of the spherical coordinate system imposed at p . See Figs. 4 and 3 for demonstration of this operation.

4.1.2. Sphere-to-sphere intersection

For pruning the candidates we chose a greedy-iterative scheme, which discards the weaker candidates according to the saliency metric (see Fig. 6). This scheme was preferred over other possibilities for several reasons. First, for performance

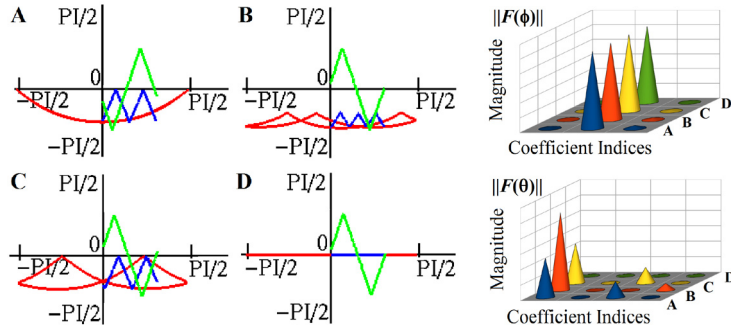


Fig. 7. Transformation of intersection curve to SID. A, B, C, D here correspond to A, B, C, D in Fig. 4. (Left) red: intersection curve projection to spherical coordinates as $(\phi(t), \theta(t))$, the curve is plotted by the following equations $x = \phi(t)$ and $y = \theta(t)$. Green: $\phi(t)$ component of parametric equation $(\phi(t), \theta(t))$, the function is plotted by the following equations $x = t$ and $y = \phi(t)$. Blue: $\theta(t)$ component of parametric equation $(\phi(t), \theta(t))$, the function is plotted by the following equations $x = t$ and $y = \theta(t)$. (Upper right) $\|\mathcal{F}(\phi)\|$ of A, B, C, D features. (Lower right) $\|\mathcal{F}(\theta)\|$ of A, B, C, D features. Note that $\|\mathcal{F}(\phi_A)\| \approx \|\mathcal{F}(\phi_C)\|$ and $\|\mathcal{F}(\theta_A)\| \approx \|\mathcal{F}(\theta_C)\|$ even though $(\phi_A(t), \theta_A(t)) \neq (\phi_C(t), \theta_C(t))$. This is a demonstration of SIDs rotation invariance. Also, note the similarity of all $\phi(t)$, its value varies on rare and complex features, thus this component provides less discriminative information compared to $\theta(t)$. For this reason we usually used fewer coefficients of $\|\mathcal{F}(\phi(t))\|$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and simplicity. Second, the focus of this work is to explore the sphere intersection method as a feature descriptor and leave further research on more complex detection schemes to future work.

Since the surface of the local neighborhood of a vertex is sampled at several radii around the vertex, we argue that it is sufficient to have at most one sphere intersecting a face in order to achieve good description of the entire shape. This reduces the problem to eliminating weaker candidates when sphere-to-sphere intersections occur.

We perform the sphere-to-sphere intersection detection by keeping track of the faces inside each sphere, thus the detection is made very efficient by simply querying the faces that are intersected or inside of more than one sphere. Furthermore, the candidates are sorted before the test in descending order by the saliency metric, this ensures that no candidate is unnecessarily eliminated.

4.1.3. Saliency metric

We follow the observation of Marr (1982) which correlates between “interesting shape features” and surface sharpness. We define interesting points as points having non-smooth or flat curvature. The detection of feature points is crucial: important parts of the objects should not be missed, while the overall number of feature points directly affects the computational complexity and introduces unimportant information.

Since our feature is an intersection curve along the mesh surface, many formulations can be considered as sole saliency metric or as a combined weighted sum – *integral*, *Laplacian* and *variance* just to name a few. For our implementation we chose to use a weighted sum of *integral of absolute value of the intersection curve* and *sum of Laplacians*. Given a vertex v let $\theta_v(t)$, $\phi_v(t)$ be defined for $t \in 1, \dots, n$ then the *sum of Laplacians* of v is defined by:

$$l_v = \sum \frac{\partial^2 \theta_v}{\partial t^2} \quad (4)$$

and *integral of absolute value of the intersection curve* of v is defined by:

$$i_v = \sum_{t=1}^{n-1} \left\| \frac{(\theta_v(t+1) - \theta_v(t)) \cdot (\phi_v(t+1) + \phi_v(t))}{2} \right\| \quad (5)$$

then the saliency metric is given by a normalized weighted sum of l_v and i_v .

This formulation gives high saliency values both to features at local extrema/minima and also to features with great variation in their local neighborhood. In Fig. 2 the left feature is considered strong because of its *sum of Laplacians* value while the feature on the right is considered strong because of high *integral of absolute value of the intersection curve* value.

In our implementation we set $n = 128$ and Laplacian aperture size to be 15. The weight of the components varies according to input characteristics. For example, for meshes of living creatures with limbs and edges of varying properties (see Fig. 12) only *integral of absolute value of the intersection curve* was used. Alternatively, for simple geometric meshes a weighted sum of *integral of absolute value of the intersection curve* and *sum of Laplacians* was used (see Fig. 1 and Fig. 11).

4.2. Feature description

Since the intersection curve may or may not be a function in spherical coordinates, we first parameterize the curve to its components $\phi(t)$ and $\theta(t)$. Each component separately is a function, so we then transform ϕ and θ curve functions

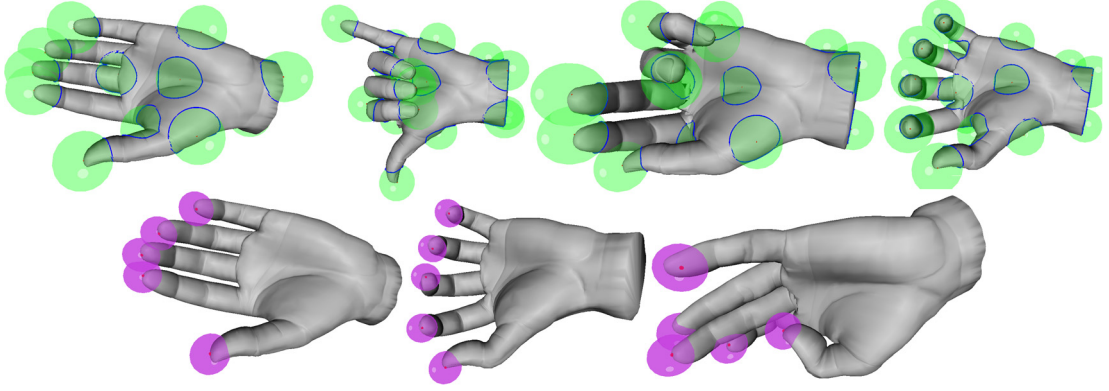


Fig. 8. Up: Detected features and their supporting spheres on hands db from Hassner and Basri (2013). Bottom: class of fingertips features as detected applying simple Bag-of-Features clustering on the resulting features.

into frequency domain using DFT and consider only coefficient magnitudes as they carry most of the information. Thus, by considering magnitudes, our descriptor becomes rotation invariant. Furthermore, we utilize the iterative nature of the feature detection phase to consider this formulation at multiple scales $[r_1 - r_{max}]$ thus providing a discrete sampling of co-centric intersection curves around a keypoint (see Fig. 3).

For a given vertex v we consider the following as a feature descriptor:

$$SID(v_i) = \|\mathcal{F}(\phi_v^{r_1})\|, \|\mathcal{F}(\theta_v^{r_1})\|, \dots, \|\mathcal{F}(\phi_v^{r_{max}})\|, \|\mathcal{F}(\theta_v^{r_{max}})\| \quad (6)$$

This descriptor is computed for each keypoint detected in the feature detection phase.

An additional advantage of using Fourier Transform is the control it gives us on the desired level of detail. We can take few coefficients from the result of the DFT to favor compactness or many coefficients to favor descriptiveness. We empirically chose to take more $\|\mathcal{F}(\theta)\|$ coefficients since it carries more information than $\|\mathcal{F}(\phi)\|$.

4.2.1. Rotation invariance

In order to underline and clarify the rotation invariance of our descriptor we provide the following elaboration using both a figure and a formal discussion.

Fig. 4 and Fig. 7 demonstrate the rotational invariance. Features A and C in Fig. 4 are both similar in shape since they are both on a straight angle edge, but those features differ in rotation and in phase. The phase difference is observable in Fig. 4 in the graphs of A and C. Despite the rotation and phase shift between A and C, Fig. 7 clearly shows the final descriptor is similar ($\|\mathcal{F}(\phi_A)\| \approx \|\mathcal{F}(\phi_C)\|$ and $\|\mathcal{F}(\theta_A)\| \approx \|\mathcal{F}(\theta_C)\|$) both in the ϕ (Fig. 7, upper right, A and C rows) and θ (Fig. 7, upper right, A and C rows) components.

For the general case, consider a surface S_1 , a point v_1 on S_1 , a surface $S_2 = R(S_1)$ where R is rotational transformation and a point $v_2 = R(v_1)$, notice that v_2 is on S_2 . For a given radius r we set a sphere around v_1 and set another sphere with radius r around v_2 . Next, let i_1 be the intersection curve of the first sphere with S_1 , and correspondingly let i_2 be the intersection curve of the second sphere with S_2 . The *consistent normal* (see Section 4.1.1 for definition) W_1 is the vector between v_1 and the centroid of i_1 , W_2 is constructed similarly using v_2 and i_2 . Notice that $W_2 = R(W_1)$. Next, U_1V_1 are the arbitrary, orthonormal axes to W_1 axis. U_2V_2 are the equivalent with respect to W_2 . Denote ϕ_1 and θ_1 as i_1 transformed to spherical coordinates and broken down to components according to the $U_1V_1W_1$ coordinate system. Denote ϕ_2 and θ_2 as i_2 transformed to spherical coordinates and broken down to components according to the $U_2V_2W_2$ coordinate system. Since $W_2 = R(W_1)$, we get similar but phase shifted functions: $\phi_1(t) = \phi_2(t - \delta_1)$, $\theta_1(t) = \theta_2(t - \delta_2)$. Performing DFT on each component: $\phi_1, \phi_2, \theta_1, \theta_2$ and keeping only the magnitudes component eliminates exactly the phase shift. $\|\mathcal{F}(\phi_1)\| = \|\mathcal{F}(\phi_2)\|$, $\|\mathcal{F}(\theta_1)\| = \|\mathcal{F}(\theta_2)\|$. Thus, using $\|\mathcal{F}(\phi)\|$ and $\|\mathcal{F}(\theta)\|$ as feature descriptors makes invariant feature descriptor.

4.3. Shape similarity

We reduce shape similarity problem to the problem of finding a minimal cost matching in a bipartite graph. Given two shapes m_a, m_b and F_a, F_b are their detected features, we define the bipartite graph as the two sets (F_a, F_b) , whereas edge costs are computed by the L_1 distance of the magnitudes vector of features. The resulting minimal cost matching gives the best possible pairing of features, thus providing accurate similarity metric for shape matching.

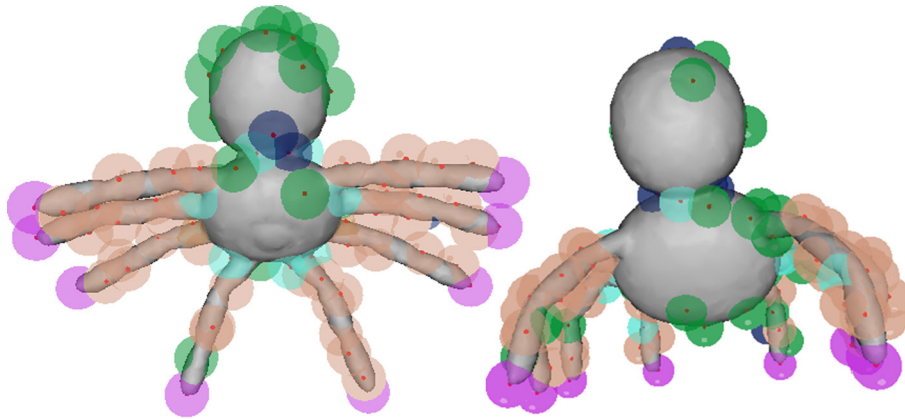


Fig. 9. Detected SIDs on spider undergoing a non-rigid deformation. Here matching descriptors are colored the same. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

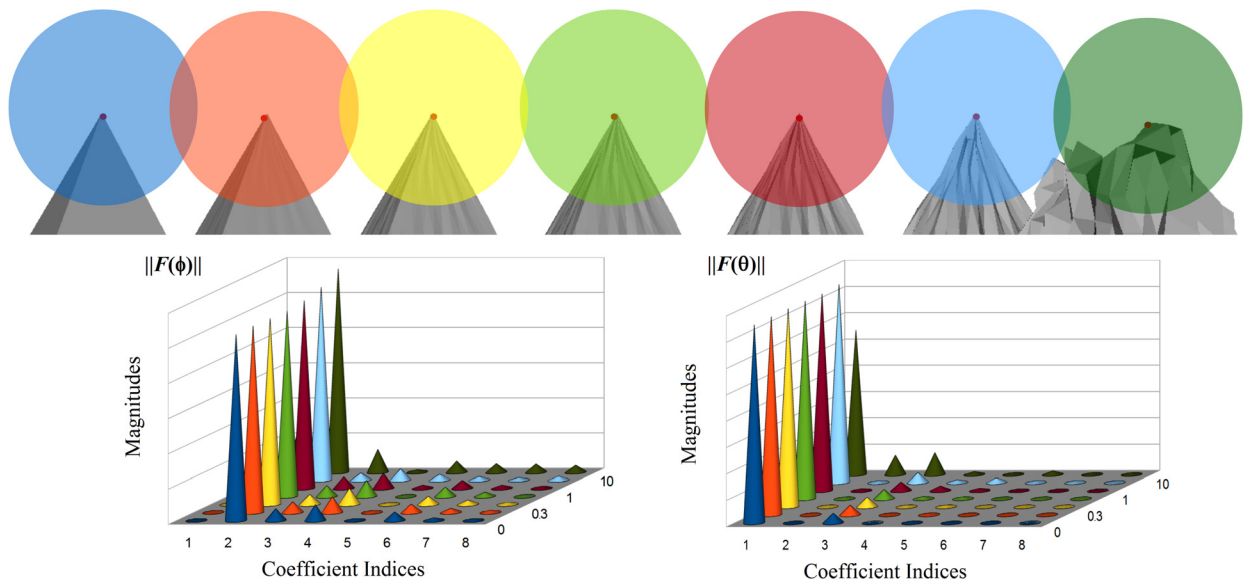


Fig. 10. Robustness to noise evaluation. Up: pyramid tip with rising degree of noise and intersecting sphere. The noise follows gaussian distribution in the normal direction and is constrained to the following percentages of the models bounding sphere radius (left to right): 0%, 0.1%, 0.3%, 0.5%, 1%, 2%, 10%. Lower left: $\|F(\phi)\|$ of all feature. Lower right: $\|F(\theta)\|$ of all feature.

5. Results and details

Our implementation is written in C++ using OpenMesh (Bischoff et al., 2002), OpenCV (Bradski, 2000) and LEMON (Dezső et al., 2011) for min cost bipartite graph matching. We make our source code available for public to encourage experimentation and further exploration of the SID shape descriptor, <http://sourceforge.net/projects/sphere-intersection-signature/>.

We have tested our implementation on various polygonal models. Figs. 1, 8 and 9 present a qualitative demonstration of our descriptor effectiveness where similar features are accurately matched together (i.e. colored by the same color).

To further evaluate our descriptors robustness against noise we performed the evaluation presented in Fig. 10. This qualitative evaluation demonstrates that our descriptor handles noise well.

We experimented with different benchmarks in order to quantify the ability of our approach to measure similarity among various models and its robustness against various transformation types. To measure the robustness against non-isometries and noise we used a subset of the SHREC 12 – stability on abstract shapes benchmark (Biasotti et al., 2012). Fig. 11 shows that our method has performed successfully on this subset.

To thoroughly evaluate our descriptor against non-rigid deformations we have experimented with the SHREC 11 non-rigid benchmark (Boyer et al., 2011). In Fig. 12 we show a subset of our results on this database. The configuration used is presented in Table 1.

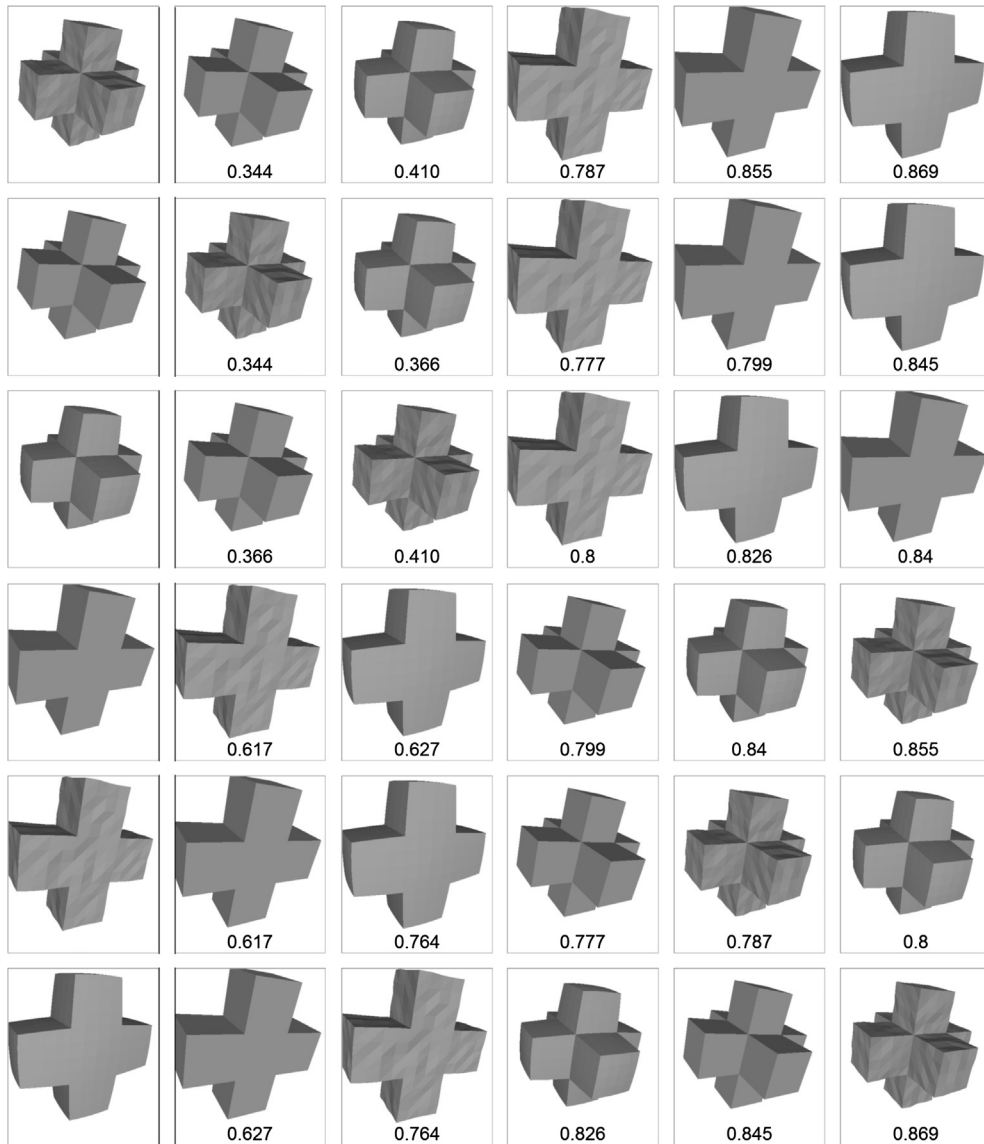


Fig. 11. Result of our method on a subset of the SHREC 12 stability on abstract shapes database. For each row, left is the query model then five results from best to worst, with similarity score beneath. This table demonstrates handling with noise and non-isometries.

Table 1

Parameter setting used for the SHREC 11 non-rigid evaluation.

Parameter	Value
Δr	1% mesh size
Iterations	10
Radii taken for descriptor	4%, 7%, 10% mesh size
Max count of features per mesh	30
<i>Integral of absolute value of the intersection curve weight</i>	1
Number of coefficients used from $\ \mathcal{F}(\phi)\ $	2
Number of coefficients used from $\ \mathcal{F}(\theta)\ $	4

Note that for this test we only used the *integral of absolute value of the intersection curve* since this DB consists mainly of animals with sharp features such as limbs. We perform our experiments on a PC machine with 2 GHz dual core 32bit CPU and 2 GB RAM. We compute shape features in 5 seconds on average for 9.5 K vertex meshes, comparing 2 models takes 14 milliseconds on average.

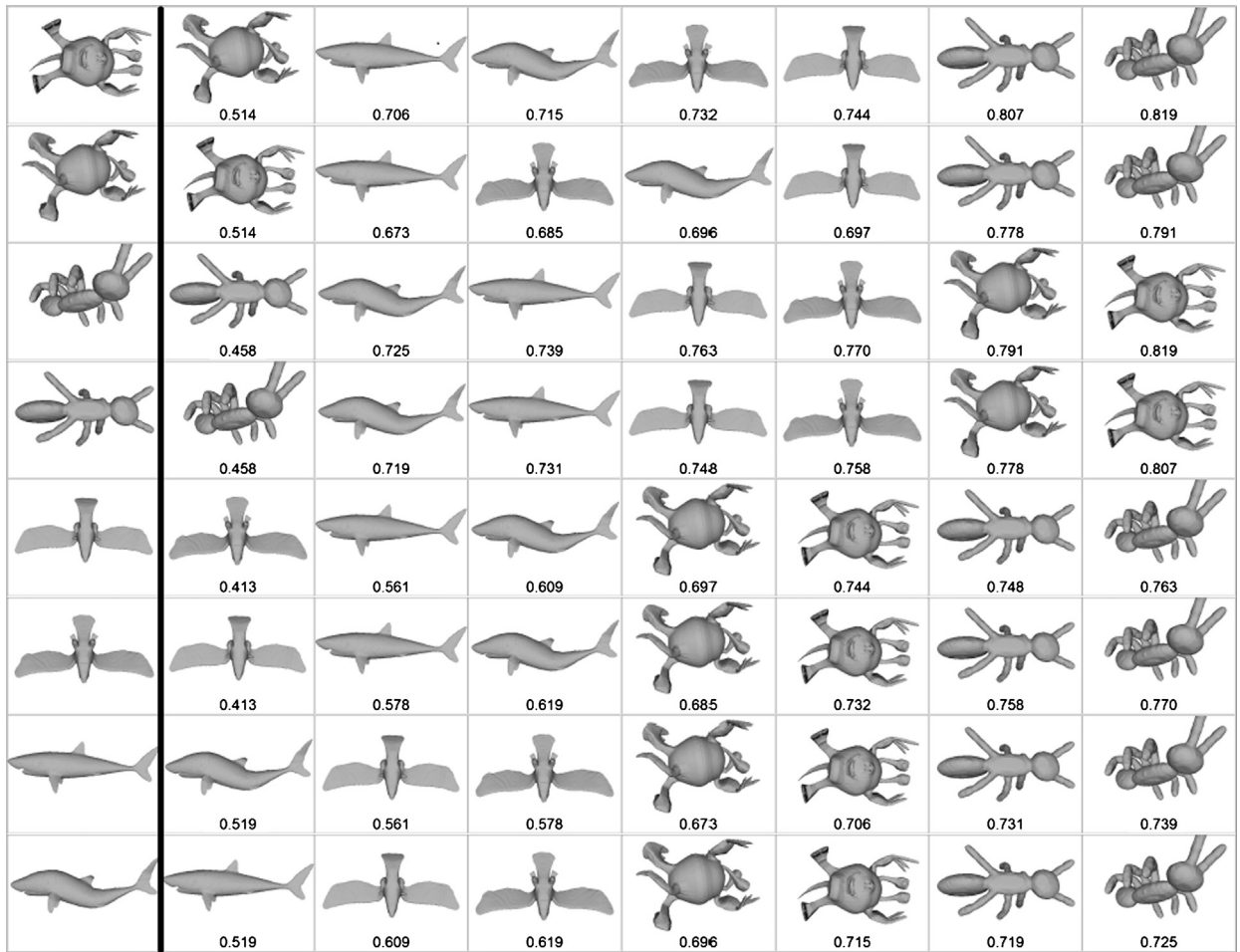


Fig. 12. Subset of our results on the SHREC 11 non-rigid benchmark. For each row, left is the query model then five selected results from best to worst, with similarity score beneath.

Table 2

Performance evaluation on SHREC 11 – non-rigid. Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), E-measure (E), and Discounted Cumulative Gain (DCG). See PSB (Shilane et al., 2004) for full definition of the measures.

Method	NN	FT	ST	E	DCG
SID (Our)	0.973	0.767	0.878	0.638	0.934
HKS	0.837	0.406	0.497	0.353	0.730
3D-Harris	0.562	0.325	0.466	0.322	0.654
MeshSIFT	0.995	0.884	0.962	0.708	0.980

5.1. Comparisons with other methods

In Fig. 13 and Table 2 we provide a comparison of our method with state-of-the-art techniques running on the SHREC 11 non-rigid benchmark. HKS and Harris-3D are shown as examples for local feature detectors that extract meaningful interest points, as demonstrated in Tombari et al. (2013), and MeshSIFT is shown as an example for state-of-the-art local feature descriptor with high performance results.

We discuss our comparison with respect to each method in detail:

Harris3D (Sipiran and Bustos, 2011) In SHREC11 (Boyer et al., 2011) the authors of Harris3D chose to use a geodesic map scheme. After detecting interest points by the Harris3D method, a geodesic map is constructed by considering normalized geodesic distances between all pairs of interest points. In contrast to considering mere scalar distances between keypoints, SID considers the geometry of the features themselves. We attribute the superiority of SID on Harris3D to the usage of fine feature geometry rather than distances.

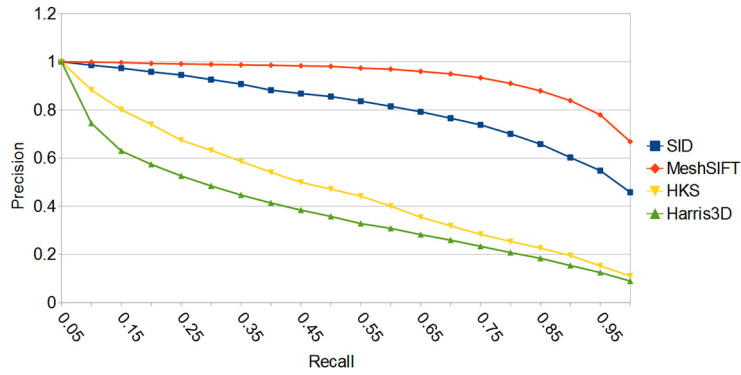


Fig. 13. Precision-recall graph on the SHREC11-non-rigid benchmark. See PSB (Shilane et al., 2004) for full definition of the precision-recall measure.

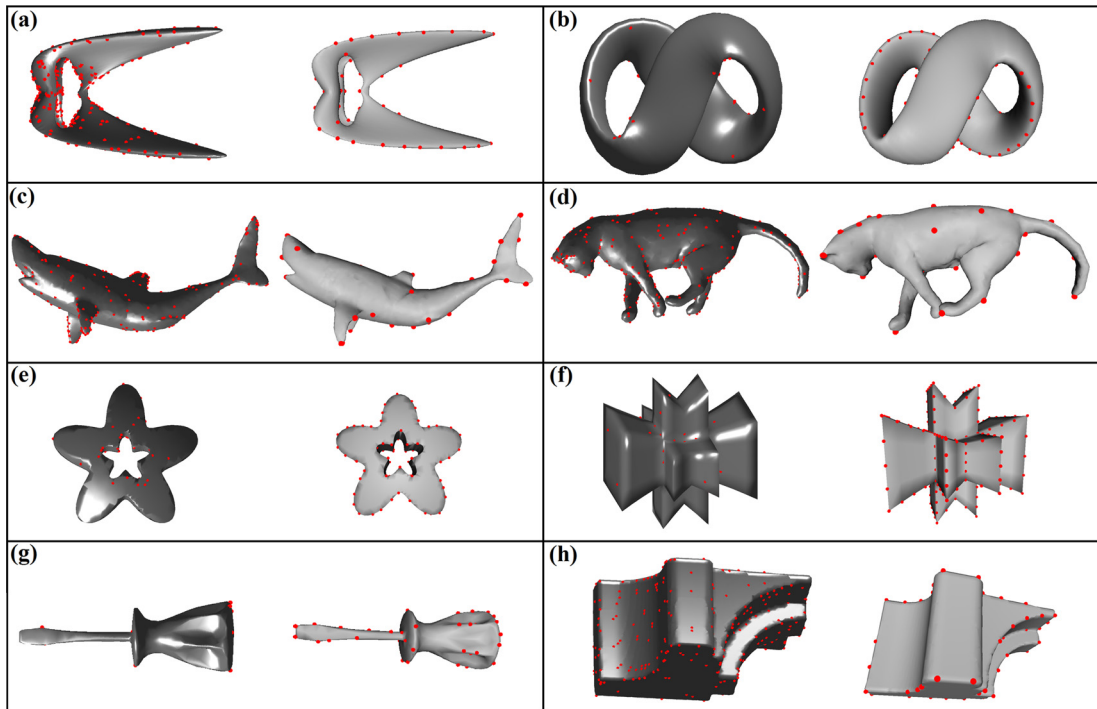


Fig. 14. Comparison of feature detection results of MeshSIFT (first and third columns, dark models) and SID (second and fourth columns, bright models). The test models are of various classes – abstract shapes with round curves (a, b), biological models (c, d), abstract shapes with straight angles (e, f), and CAD models (g, h). On all classes SID performs better than MeshSIFT by choosing features that are more meaningful, salient and symmetric.

HKS (Sun et al., 2009) based point-to-point matching HKS uses a descriptor comprised of samples along the continuous representation of the Heat Kernel Signature. SID also uses discrete sampling of functions ($\phi(t), \theta(t)$), but it compacts the descriptor even further by performing DFT. Thus, HKS takes a descriptor of size 100 whereas SID takes a descriptor of size 18 ($(2\phi$ coefficients + 4θ coefficients) $\times 3$ radii taken). Using a descriptor of a larger size can indicate “unimportant” information carried in each descriptor, thus making the comparison between features more difficult. We suggest that SID outperforms HKS due to the descriptors distilled feature representation.

MeshSIFT (Maes et al., 2010) While SID intersects the surface at predefined radii, MeshSIFT uses the entire local neighborhood of a keypoint for computing the descriptor. Thus, MeshSIFT captures the entire area of the surface. It is up to the user of SID to manually choose correctly the predefined radii of the spheres, otherwise, some area of the shape might not be sampled.

In order to further explore the differences between SID and MeshSIFT we perform a comparison of features detected by each method on various classes of models (see Fig. 14). We used MeshSIFT (Smeets et al., 2013) source code supplied by the original authors with default parameter setting (other settings were experimented as well, but yielded poorer results). For the comparison we use several class of models – abstract shapes with round curves (a, b), biological models (c, d), abstract shapes with straight angles (e, f), and CAD models (g, h). MeshSIFT results are depicted in first and third columns

Table 3

Comparison of runtimes on test models from Fig. 14. The numbers are given in seconds.

	MeshSIFT	SID
A	2635	18
B	92	5
C	448	7
D	869	9
E	46	1
F	22	1
G	7	0.5
H	1174	16
Average	661	7

as dark models, while SID results are depicted in second and forth columns as bright models. Although both methods are not symmetry-aware, SID detects same features on models with self symmetry while MeshSIFT does not, which makes SID more predictable and reproducible. This is particularly emphasized in (a), (e) and (f). Furthermore, SID chooses meaningful keypoints that are salient and generally not flat while MeshSIFT detects both sharp and flat features, this is particularly emphasized in (c) and (h). Another interesting observation is that on some models MeshSIFT can choose very few keypoints which may lead to sub-optimal description of a mesh, this is particularly emphasized in (b) and (g).

Another important advantage of SID over MeshSIFT are runtimes, Table 3 illustrates that SID is up to 100x faster than MeshSIFT.

To conclude our comparison with MeshSIFT we suggest the following, SID offers benefits over MeshSIFT such as speed, meaningful keypoint selection and consistency of keypoint selection. Nevertheless, we observed a slightly better performance of MeshSIFT over SID in a specific class of models (i.e, SHREC 11 models are biological with non-rigid transformations) where parameters may be carefully tuned.

6. Limitations and conclusions

Limitations of our method include taking only a single intersection curve even if the sphere-to-surface intersection results in several curves. We limit the negative effect of this limitation by choosing this curve consistently by choosing a small enough initial sphere radius and Δr (see Section 4.1.1). As an evidence, in all of our experiments we did not encounter a jitter of the intersection curve, namely, a situation in which the chosen intersection curve “jumps” between several possible intersection curves.

A further limitation is that feature matching false positives are possible, since we consider only the magnitudes and not the phases of the transformation of the intersection curve. However, we chose to use only the magnitudes since we aimed to minimize dimensionality of the descriptor and we observed empirically that the magnitude carry most of the important information.

Additional limitation of our implementation is support for watertight triangular meshes only.

We have presented a novel shape descriptor for efficient matching in large 3D databases and reported experimental analysis for its performance on various datasets. Our shape descriptor is highly distinctive, as it captures the entire (2π) relative curvature of the neighborhood of a point. It allows even a single feature to find a correct match with good probability in a large set of features. Our shape descriptor is compact as it represents the intersection of a sphere with a surface as 2 1D curves. The sphere intersection operation is simple and relies only on the point position, thus can be defined robustly everywhere on a surface. The use of a 2D image based representation allows efficient processing using image processing algorithms. Our experimental results show that our descriptor manages to capture the similarity among various dataset and is robust against various types of deformations.

The scope of future work includes exploring additional more feature detection schemes, incorporating different techniques that capture the topological relation among the features of the model to the graph matching, enhancing our method with smart selection of radii to achieve scale invariance, performing soft clustering for use of bag of features approach, handling many intersection curves for each sphere, extending the presented descriptor to handle non-watertight and point cloud datasets and evaluating the descriptor with respect to new applications.

Acknowledgements

This work was supported by grants from the Lynn and William Frankel Center for Computer Sciences, Ben-Gurion University, Israel 428/11, as well as a grant from the Israeli Science Foundation. The authors would also like to thank the anonymous reviewers for the helpful suggestions and comments.

References

- Ankerst, M., Kastenmüller, G., Kriegel, H.P., Seidl, T., 1999. 3D shape histograms for similarity search and classification in spatial databases. In: Advances in Spatial Databases. In: 6th International Symposium, SSD'99, vol. 1651, pp. 207–228. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.9487>.

- Arbter, K., Snyder, W.E., Burhardt, H., Hirzinger, G., 1990. Application of affine-invariant Fourier descriptors to recognition of 3-d objects. *IEEE Trans. Pattern Anal. Mach. Intell.* 12, 640–647. <http://dx.doi.org/10.1109/34.56206>.
- Arkin, E.M., Chew, L.P., Huttenlocher, D.P., Kedem, K., Mitchell, J.S.B., 1990. An efficiently computable metric for comparing polygonal shapes, pp. 129–137. <http://dl.acm.org/citation.cfm?id=320176.320190>.
- Belongie, S., Malik, J., Puzicha, J., 2001. *Matching Shapes*, vol. 1, pp. 454–461.
- Biasotti, S., Bai, X., Bustos, B., Cerri, A., Giorgi, D., Li, L., Mortara, M., Sipiran, I., Zhang, S., Spagnuolo, M., 2012. Shrec'12 Track: Stability on Abstract Shapes, pp. 101–107.
- Bischoff, S., Botoch, M., Steinberg, S., Kobbelt, L., 2002. Openmesh – a Generic and Efficient Polygon Mesh Data Structure.
- Boyer, E., Bronstein, A.M., Bronstein, M.M., Bustos, B., Darom, T., Horaud, R., Hotz, I., Keller, Y., Keustermans, J., Kovnatsky, A., Litman, R., Reininghaus, J., Sipiran, I., Smeets, D., Suetens, P., Vandermeulen, D., Zaharescu, A., Zobel, V., 2011. Shrec 2011: Robust Feature Detection and Description Benchmark, pp. 71–78. <http://dx.doi.org/10.2312/3DOR/3DOR11/071-078>.
- Bradski, G., 2000. The openCV library. *Dr. Dobb's Journal of Software Tools*.
- Bronstein, M.M., Kokkinos, I., 2010. Scale-Invariant Heat Kernel Signatures for Non-rigid Shape Recognition, pp. 1704–1711.
- Castellani, U., Cristani, M., Fantoni, S., Murino, V., 2008. Sparse Points Matching by Combining 3d Mesh Saliency with Statistical Descriptors, vol. 27, pp. 643–652.
- Castelli, V., Bergman, L., 2001. *Image Databases: Search and Retrieval of Digital Imagery*.
- Chua, C.-S., Jarvis, R., 1996. 3d free-form surface registration and object recognition. *Int. J. Comput. Vis.* 17 (1), 77–99.
- Chua, C.S., Jarvis, R., 1997. Point signatures: a new representation for 3d object recognition. *Int. J. Comput. Vis.* 25, 63–85.
- Darom, T., Keller, Y., 2012. Scale-invariant features for 3-d mesh models. *IEEE Trans. Image Process.* 21 (5), 2758–2769.
- Delingette, H., Hebert, M., Ikeuchi, K., 1993. A spherical representation for the recognition of curved objects. In: *Proceedings of the Fourth International Conference on Computer Vision*, 1993, pp. 103–112.
- Dezső, B., Jüttner, A., Kovács, P., 2011. Lemon – an open source C++ graph template library. *Electron. Notes Theor. Comput. Sci.* 264 (5), 23–45.
- Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P., 1995. Query by image and video content: the qbic system. *IEEE Comput.* 28, 23–32.
- Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., Jacobs, D., 2003. A search engine for 3d models. *ACM Trans. Graph.* 22, 83–105. <http://dx.doi.org/10.1145/588272.588279>.
- Giorgi, D., Mortara, M., Spagnuolo, M., 2010. 3d shape retrieval based on best view selection, pp. 9–14. <http://doi.acm.org/10.1145/1877808.1877812>.
- Hassner, T., Basri, R., 2013. Single view depth estimation from examples. *arXiv preprint arXiv:1304.3915*.
- Horn, B.K.P., 1984. Extended Gaussian images. *Proc. IEEE* 72 (12), 1671–1686.
- Jacobs, C.E., Finkelstein, A., Salesin, D.H., 1995. Fast Multiresolution Image Querying, pp. 277–286.
- Johnson, A.E., Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* 21, 433–449.
- Kashyap, R.L., Chellappa, R., 1981. Stochastic models for closed boundary analysis: representation and reconstruction. *IEEE Trans. Inf. Theory* 27 (5), 627–637.
- Kazhdan, M., Funkhouser, T., Rusinkiewicz, S., 2003. Rotation Invariant Spherical Harmonic Representation of 3d Shape Descriptors, pp. 156–164. <http://dl.acm.org/citation.cfm?id=882370.882392>.
- Kokkinos, I., Bronstein, M.M., Litman, R., Bronstein, A.M., 2012. Intrinsic Shape Context Descriptors for Deformable Shapes, pp. 159–166.
- Li, X., Guskov, I., 2005. Multi-scale features for approximate alignment of point-based surfaces. <http://dl.acm.org/citation.cfm?id=1281920.1281955>.
- Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 60 (2), 91–110. <http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94>.
- Maes, C., Fabry, T., Keustermans, J., Smeets, D., Suetens, P., Vandermeulen, D., 2010. Feature Detection on 3d Face Surfaces for Pose Normalisation and Recognition, pp. 1–6.
- Marr, D., 1982. *Vision: a computational investigation into the human representation and processing of visual information*.
- Mian, A., Bennamoun, M., Owens, R., 2010. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *Int. J. Comput. Vis.* 89 (2–3), 348–361.
- Mortara, M., Patané, G., Spagnuolo, M., Falcidieno, B., Rossignac, J., 2004. Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. *Algorithmica* 38, 227–248.
- Ogle, V.E., Stonebraker, M., 1995. Chabot: retrieval from a relational database of images. *IEEE Comput.* 28, 40–48.
- Ohbuchi, R., Nakazawa, M., Takei, T., 2003. Retrieving 3d Shapes Based on Their Appearance, pp. 39–45. <http://doi.acm.org/10.1145/973264.973272>.
- Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D., 2001. Matching 3d models with shape distributions, p. 154. <http://dl.acm.org/citation.cfm?id=882486.884103>.
- Pottmann, H., Wallner, J., Huang, Q.-X., Yang, Y.-L., 2009. Integral invariants for robust geometry processing. *Comput. Aided Geom. Des.* 26 (1), 37–60. <http://dx.doi.org/10.1016/j.cagd.2008.01.002>.
- Shilane, P., Min, P., Kazhdan, M., Funkhouser, T., 2004. The Princeton Shape Benchmark, pp. 167–178.
- Sipiran, I., Bustos, B., 2011. Harris 3d: a robust extension of the Harris operator for interest point detection on 3d meshes. *Vis. Comput.* 27 (11), 963–976.
- Smeets, D., Keustermans, J., Vandermeulen, D., Suetens, P., 2013. Meshsift: local surface features for 3d face recognition under expression variations and partial data. *Comput. Vis. Image Underst.* 117 (2), 158–169. <http://dx.doi.org/10.1016/j.cviu.2012.10.002>.
- Stein, F., Medioni, G., 1992. Structural indexing: efficient 3-D object recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (2), 125–145. <http://dx.doi.org/10.1109/34.121785>.
- Sun, J., Ovsjanikov, M., Guibas, L., 2009. A concise and provably informative multi-scale signature based on heat diffusion, pp. 1383–1392. <http://dl.acm.org/citation.cfm?id=1735603.1735621>.
- Tombari, F., Salti, S., Di Stefano, L., 2013. Performance evaluation of 3d keypoint detectors. *Int. J. Comput. Vis.* 102 (1–3), 198–220.
- Veltkamp, R.C., Hagedoorn, M., 2001. State of the Art in Shape Matching, pp. 87–119. <http://dl.acm.org/citation.cfm?id=370792.370810>.
- Zaharescu, A., Boyer, E., Varanasi, K., Horaud, R., 2009. Surface Feature Detection and Description with Applications to Mesh Matching.
- Zhang, D., Hebert, M., 1999. *Harmonic Maps and Their Applications in Surface Matching*, vol. 2.