

Template Based Shape Descriptor

Raif M. Rustamov

Drew University, Madison, NJ, USA

Abstract

We introduce a new 3D shape descriptor which maps the surface features onto an arbitrary template surface using mean-value interpolation. A compact numerical shape descriptor is extracted using manifold harmonics on the template. We show that mean-value interpolation is a strong alternative to the often used projection. The utility of using different templates is established by showing that concatenating descriptors coming from different templates improves retrieval quality.

1. Introduction

Content based 3D shape retrieval has drawn considerable attention of researchers. Existing surveys [IJL*05, BKS*05, TV08, SB07, BDFF*08] not only describe, compare, and test the proposed approaches but also introduce their classifications. Without trying to provide another comprehensive classification method nor claiming any novelty, let us set up the context for our approach by highlighting three components that frequently go into the construction of some numerical shape descriptors. The discussion afterward should clarify our use of this specific component delineation and naming.

The first component is the *selection* of surface feature. By surface feature we mean any function on the surface that captures a property relevant to shape description. Examples include the constant function (restriction of the surface's characteristic function to the surface itself), distance to the center of mass, curvature, components of the normal vector, and many others. We refer to the selected function as the feature function.

The second component is *mapping* the feature function to, perhaps, a different domain. The feature function is used to construct a new function defined on some predetermined domain; we call this domain as the mapping domain, and the new function as the mapped feature function. The existing approaches use spheres, planes, the 3D space (surface's bounding volume), the surface itself as mapping domains. The mapped feature function is usually constructed by some kind of projection. When the mapping domain is the surface itself, the mapped feature function is just set equal to the feature function.

The third component is *signal processing* to extract a concise, noise-robust numerical shape descriptor from the mapped feature function. The type of processing depends on the mapping domain: for sphere one may use the Spherical Harmonic Transform, for a plane or a box volume – 2D or 3D Fourier transform, ball volume – 3D Zernike Transform. Essentially, in all of these cases the mapped feature function is expanded in a series in terms of the relevant basis, and the expansion coefficients are used as the shape descriptor.

To provide examples, let us analyze a few existing descriptors in the described terms. Consider the descriptor of [SV01] where rays are shoot from the origin (center of mass of the mesh) to determine the distance to the farthest intersection point with the mesh. The rays can be parameterized by the unit sphere, and as a result one obtains a function defined on the sphere. Next, the spherical harmonic transform is applied to this function to extract the numerical shape descriptor. In our terms, the feature function is the distance from the mesh point to the origin; the mapping domain is the unit sphere; mapped feature function is obtained by projecting the feature function onto the sphere, and if two mesh points project to the same sphere point, one resolves the ambiguity by selecting the larger function value; finally, the signal processing component is performed using the spherical harmonic transform.

The depth-buffer descriptor [HKSV02] places a normalized mesh into a unit cube, and generates six gray-scale images on each face of the cube by parallel projection. The grayness value is set based on the distance from the cube face to the model. Coefficients obtained by applying the 2D Fourier transform to each of the six gray-scale images

are used as descriptors. In our terms, for each of the cube faces, the feature function is the distance from mesh point to the face; mapping domain is the cube face; the mapped feature function is obtained by projecting the feature function onto the face, and if two mesh points project to the same face point, one resolves the ambiguity by selecting the smaller function value; the signal processing component is performed using the 2D Fourier transform.

These examples can be multiplied especially by noting that our second and third components roughly correspond to the object abstraction and numerical transformation components of [BKS*05]. After such a consideration the following picture emerges: numerical descriptors tend to use a mapping domain different from the original surface, and in almost all of such cases the mapping domain is one of the primitive geometries.

This observation is not surprising. In fact, when the mapping domain is not a primitive geometry one is limited to using a histogram, and histograms, except when a few judiciously chosen features are combined to obtain a joint distribution [ASSY07], do not capture as much information as the expansion coefficients obtained using signal processing. Indeed, expansion coefficients can be used to reconstruct the original function up to some precision. Thus, one is led to use mapping domains that are equipped with well established signal processing tools, which limits the possible choices to simple geometries.

However, it was suggested by Mademlis et. al. [MDTS08b] that since ellipses approximate elongated shapes better than spheres, they were more appropriate to serve as mapping domains than spheres. Indeed, Mademlis et. al. showed experimentally that using ellipsoidal harmonics gives better retrieval results than the spherical harmonics of [KFR03].

Our main contribution is to take this suggestion of Mademlis et. al. further by allowing the use of any fixed surface – ‘template’ – as the mapping domain. We achieve this using two ideas: first, to obtain the mapped feature function we propose to replace projection by mean-value interpolation; second, we use the Laplace-Beltrami eigenfunctions – manifold harmonics in the terminology of [VL08] – to perform signal processing on the mapping domain. In other words, interpolation is applied to extend the feature function from the model surface to the template surface, then, we expand the new function in terms of the template’s manifold harmonics, consequently extracting the low-frequency expansion coefficients as the shape descriptor.

It may be tempting to altogether bypass mapping to a fixed template, and expand the feature function in terms of the manifold harmonics of the original surface. However, it is very difficult to match the manifold harmonics coming from different surfaces; see [JZ06] for problems encountered when dealing with eigenfunctions. The use of a fixed template avoids this problem by making sure that the

extracted expansion coefficients are in correspondence and their comparison is meaningful. Let us note, however, that it is possible to match some other spectral properties directly, and there exist shape descriptors based on the eigenvalues [RWP06] and eigenfunction distance histograms [Rus07] of the Laplace-Beltrami operator.

Mean-value interpolation [Flo03,FKR05,JSW05] is a first order polynomial precision interpolation scheme that can be used to extend a piecewise linear function from a surface mesh to the entire ambient Euclidean space. Our reliance upon mean-value interpolation to obtain the mapped feature function rather than projection can be justified as follows. For projection, a correspondence between the model and template surface points is established, and the feature function values are transferred to the template using this correspondence. Since the correspondence is generally not one-to-one, the resulting mapped feature function can be discontinuous when there are overlaps. Therefore, its expansion in terms of manifold harmonics will be subject to the Gibbs effect, rendering the low-frequency expansion coefficients – the ones used as the shape descriptor – inadequate for representing the function. In addition, when different templates are used, we expect projection to lead to high redundancy, limiting the gains of concatenating descriptors obtained from different templates. Indeed, if there are no overlaps, the value sets of the mapped feature functions on various templates will be the same, with only difference being how the values are distributed over the template surfaces.

In contrast, the mean-value interpolation has the following benefits. First, the relative positions and distances between the template and the model surface influence the mean-value interpolant greatly, which leads to less redundancy between interpolants obtained for different templates. Second, the mean-value interpolant is continuous. Third, in principle, a mesh can be reconstructed if the mean-value coordinates – mean-value interpolation is based on these – are given everywhere in the space; consequently, one may reasonably expect that interpolation based on mean-value coordinates implicitly injects even more shape information into the mapped feature function.

Our experimental results confirm that mean-value interpolation provides a strong alternative to projection, and that manifold harmonic expansion coefficients are useful as shape descriptors. We also show that concatenating descriptors coming from different templates improves retrieval quality.

2. Construction

We describe the construction of the template based shape descriptor stage by stage, following the component description of Introduction. Models in the search database are assumed to be given as triangle meshes; we will use the terms surface, surface mesh, and mesh interchangeably. We let S denote any surface in the database. All models in the database

are assumed to be normalized to achieve translation, rotation and scale invariance of the resulting shape descriptor. The standard normalization is used: first, the origin is shifted to the center of mass of the surface mesh; second, continuous PCA of [VSR01] is used to orient the mesh; third, the mesh is isotropically scaled to have a unit variance. The effectiveness of such a normalization has been corroborated in the literature [BKS*05].

A fixed surface T will be used as the mapping domain; we refer to it as the template. The template can be one of the surfaces in the database, or any other closed manifold surface mesh chosen arbitrarily.

2.1. Selection

A variety of possibilities exist as a choice for the surface feature – distance to the center of mass, curvature, components of the normal vector, and many others. We denote the feature function by f , where $f : S \rightarrow R$ is a real-valued function on the model surface. Let us mention that due to the interpolation schemes employed, f cannot be the characteristic function of the surface. In fact, it restricts to a constant function on the surface, and the interpolation schemes we consider extend the constant function to a constant function everywhere, making impossible extraction of any meaningful information. Linear functions of the x, y, z coordinates cannot be used for a similar reason – they also extend to a linear function everywhere when mean-value interpolation is used. To focus our discussion, we follow [SV01] and choose the feature function to be the distance to the center of mass – the origin: $f(p) = |p|, p \in S$.

2.2. Mapping

In this step, one constructs the mapped feature function using the feature function of the surface, namely we define a new function $\hat{f} : T \rightarrow R$ based on the given function $f : S \rightarrow R$. For comparison we will consider two schemes: Shepard interpolation and mean-value interpolation.

Given a set of scattered points p_i where the values $f(p_i)$ are known, the Shepard interpolant [She68, GW78] is defined as follows

$$\hat{f}(p) = \frac{\sum_i w_i(p) f(p_i)}{\sum_i w_i(p)}, \quad p \in R^3$$

where w_i are user assigned weights. The most commonly used weights are of the following form

$$w_i(p) = |p - p_i|^{-\alpha},$$

where α is a positive parameter. Shepard interpolant has the zeroth order precision meaning that the constant functions are reproduced, i.e. if f is a constant function, then \hat{f} is also constant.

In our experiments, to obtain points p_i we uniformly sample the area of the model surface, and evaluate the feature

function at these points. The mapped feature function is evaluated at the vertices of the template mesh T using Shepard interpolation with the values of $\alpha = 1, 2, 3$.

Mean-value interpolation [Flo03, FKR05, JSW05] is an instance of more general barycentric interpolation. For an arbitrary triangulated surface mesh with vertices v_1, v_2, \dots, v_n , the functions $b_i : R^3 \rightarrow R, i = 1, \dots, n$ are called barycentric coordinates if they satisfy the following three properties: partition of unity,

$$\sum_{i=1}^n b_i(p) = 1,$$

affine combination

$$\sum_{i=1}^n b_i(p) v_i = p,$$

and Lagrange property

$$b_i(v_j) = \delta_{ij}.$$

Due to these properties, the barycentric coordinates can be used for interpolation: given function values $f(v_i)$ on the mesh vertices, one extends it to the whole space by setting for each $p \in R^3$,

$$\hat{f}(p) = \sum_{i=1}^n b_i(p) f(v_i).$$

The barycentric interpolant has the first order precision meaning that the linear functions of coordinates x, y, z are reproduced, i.e. if f is a linear function, then \hat{f} is also linear.

In our experiments, we use mean-value coordinates because they are fastest to evaluate. We compute the feature function values at the vertices of the model mesh, and then evaluate the mapped feature function at the vertices of the template mesh using mean-value interpolation. Our implementation of mean-value coordinates follows the pseudocode presented in [JSW05].

For both barycentric and Shepard interpolation, the relative positions and distances between the template and the model surface influence the interpolant greatly, leading to less redundancy between interpolants obtained for different templates. It is also important to notice that due to the Lagrange property, one can reconstruct the mesh vertices if the barycentric coordinates are given everywhere. In addition, the triangle faces can be reconstructed by finding the triplets of non-zero barycentric coordinates such that all remaining barycentric coordinates are zero. Therefore, a triangular mesh can be reconstructed from its barycentric coordinates, which leads us to believe that interpolation done using barycentric coordinates injects even more surface information into the interpolated feature function.

2.3. Signal Processing

Signal processing is important to shape description at least due to two factors. First, for shape description it is crucial to

reduce the amount of noise so that only the essential properties of the shape are captured. Second, compact numerical shape descriptors are naturally provided by signal processing. Both goals are achieved by expanding a given function in a series and using the expansion coefficients of the low-frequency components as the shape descriptor. Keeping only the low-frequency part not only results in a compact descriptor, but also is essentially equivalent to smoothing the function – making the descriptor more robust to noise.

The relevant signal processing bases for simple geometries – plane, sphere, space – are well known and are extensively used. However, we need to perform signal processing on the template surface T that can be any closed manifold triangle mesh. As has been corroborated in the literature, most recently in [VL08], Laplace-Beltrami eigenfunctions provide the sought basis in this setting. For example, the spherical harmonics used so often in shape retrieval are nothing but the Laplace-Beltrami eigenfunctions of the sphere.

We first review the continuous setting for Laplace-Beltrami operator. For a closed compact manifold surface T , let Δ denote its Laplace-Beltrami differential operator. Consider the equation

$$\Delta\phi = \lambda\phi.$$

A scalar λ for which the equation has a nontrivial solution is called an eigenvalue of Δ ; the solution ϕ is called an eigenfunction corresponding to λ . Note that $\lambda = 0$ is always an eigenvalue – the corresponding eigenfunctions are constant functions.

The eigenvalues of the Laplace-Beltrami operator are non-negative and constitute a discrete set; we can put them into non-decreasing order: $\lambda_1 = 0 \leq \lambda_2 \leq \lambda_3 \leq \dots \leq \lambda_i \leq \dots$. The eigenfunction corresponding to λ_i , normalized to have the unit L_2 norm will be denoted by ϕ_i .

Since the Laplace-Beltrami operator is Hermitian, the eigenfunctions corresponding to its different eigenvalues are orthogonal: $\langle \phi_i, \phi_j \rangle = \int_T \phi_i \phi_j = 0$. Thus, the eigenfunctions of Laplace-Beltrami operator give an orthogonal basis for the space of functions defined on the surface. As a result, a given function g on the surface can be expanded in terms of the eigenfunctions

$$g = c_1\phi_1 + c_2\phi_2 + \dots, \quad (1)$$

where the coefficients are given by

$$c_i = \langle g, \phi_i \rangle = \int_T g\phi_i.$$

It is known that low-frequency eigenfunctions (small λ) display less oscillation on the surface. Thus, similarly to the standard Fourier analysis, truncating the series expansion results in smoothing. Put differently, the expansion coefficients corresponding to the beginning of the series capture the essential properties of the function; for a discussion see [BNS06].

Due to its relevance to the metric to be used with the shape descriptor, let us mention that if two functions g and g' have expansions $g = \sum_i c_i \phi_i$ and $g' = \sum_i c'_i \phi_i$, then the L_2 norm of the difference between g and g' can be evaluated as

$$\begin{aligned} \int_T (g - g')^2 &= \int_T \left(\sum_i (c_i - c'_i) \phi_i \right)^2 = \\ &= \sum_{i,j} (c_i - c'_i)(c_j - c'_j) \int_T \phi_i \phi_j = \\ &= \sum_{i,j} (c_i - c'_i)(c_j - c'_j) \delta_{ij} = \sum_i (c_i - c'_i)^2, \end{aligned}$$

where we have used the orthogonality of the eigenfunctions. This equality should be kept in mind to justify our later use of Euclidean distance between the feature vectors.

In the discrete setting the Laplace-Beltrami operator of a manifold triangle mesh without boundary is given by a matrix, and its eigenvectors are used to expand a discrete function given on the mesh. For a mesh function g , the value of $\Delta g(p_i)$ at the vertex i is approximated as

$$\Delta g(p_i) \approx \frac{1}{r_i} \sum_{j \in N(i)} m_{ij} (g(p_j) - g(p_i)), \quad (2)$$

where the summation is over all vertex indices j adjacent to vertex i . It is customary to refer to r_i as the *point-areas*. For the Laplacian we use, $m_{ij} \neq 0$ only if i and j are adjacent, and the equality $m_{ij} = m_{ji}$ holds for all i, j .

Let us stack up the values of the function at mesh vertices, $g(p_i)$, to construct the column-vector \vec{g} . Now, the above formula can be written as a matrix-vector multiplication $\Delta g \approx L\vec{g}$. The involved matrix L – the discrete Laplacian – has the entries as follows

$$L_{ij} = \begin{cases} \sum_k m_{ik}/r_i & \text{if } i = j, \\ -m_{ij}/r_i & \text{if } i \text{ and } j \text{ adjacent,} \\ 0 & \text{otherwise.} \end{cases}$$

Now the Laplace-Beltrami eigenvalues/eigenfunctions are replaced by the eigenvalues λ_i and eigenvectors \vec{v}_i of the matrix L .

Since the point-areas r_i associated with mesh vertices vary from vertex to vertex, the discrete Laplacian matrix L is not symmetric. However, finding L 's eigenvalues and eigenvectors in all cases can be reduced to a symmetric eigenvalue problem. In fact, consider the diagonal matrix R with $R_{ii} = r_i$; denote by M the matrix whose entries are given by $M_{ij} = m_{ij}$ if $i \neq j$, and $M_{ii} = l_{ii}r_i$. Notice that $L = R^{-1}M$. Now we can easily check that L is similar to the symmetric matrix $K = R^{-\frac{1}{2}}MR^{-\frac{1}{2}}$. Thus, the eigenvalues and eigenvectors of L can be evaluated from those of K : if \vec{w}_i is an eigenvector of K with eigenvalue λ_i , then $\vec{v}_i = R^{-\frac{1}{2}}\vec{w}_i$ is the eigenvector of L with the same eigenvalue λ_i .

Another crucial point is that the eigenvectors of L are orthogonal. However, the orthogonality here is in terms of the

R -inner (dot) product:

$$\langle \vec{v}_i, \vec{v}_j \rangle_R = \vec{v}_i^T R \vec{v}_j = \delta_{ij}.$$

Finally, the series expansion of the mesh function g , which is represented by vector \vec{g} , is given as

$$\vec{g} = \sum_i c_i \vec{v}_i, \quad c_i = \langle \vec{g}, \vec{v}_i \rangle_R = \vec{g}^T R \vec{v}_i.$$

In our experiments we use the Laplacian of [MDSB02]: m_{ij} are the cotangent weights, and points areas r_i are given in terms of vertex Voronoi areas. The template mesh is only processed once to extract its first N eigenvectors and also the diagonal matrix R – only these are needed to obtain the expansion coefficients c_i . We use the value of $N = 20$. Since the template surfaces we consider have about 500 vertices, the processing takes less than a few seconds. Storing R (approx. 500 non-zero entries) and the eigenvectors consumes about $500 \times N + 500 = 10,500$ floating numbers.

When computing the shape descriptor for the surface S , we obtain the mapped feature function $\hat{f} : T \rightarrow R$ as described in the previous subsection. We also compute the feature function of the template, f_T . The function that is expanded in terms of the eigenvectors is $g = \hat{f}/f_T$. Since N eigenvectors and the matrix R have already been computed and stored, we only need to do the vector-matrix multiplications to extract the needed coefficients $c_i, i = 1, \dots, N$, which is done extremely fast. Finally, our feature vector consists of N floating numbers, namely $FV = (c_1, c_2, \dots, c_N)$, and as explained in the continuous setting, the Euclidean distance is the appropriate measure of distance to use between feature vectors. Notice that when S and T coincide, $g = \hat{f}/f_T = 1$, and the feature vector is of the form $FV = (a, 0, \dots, 0)$. We scale all feature vectors to achieve $a = 1$.

3. Experiments and Discussion

For our experiments we use the Watertight Benchmark [GBP07] which consists of 400 closed surface models, divided into 20 equal object classes. Some of the classes include both different objects and articulations of the same object. All objects are normalized by shifting the origin to the center of mass, scaling to obtain unit variance, and rotating using PCA.

We use feature vector size of $N = 20$; similarity is measured by Euclidean distance. The templates used in the experiments, except the sphere, were chosen randomly and are depicted in Figure 1. These were obtained from models in the benchmark by decimating using QSlim [GH97]. This was done to make the computation of shape descriptors less time consuming.

Our implementation was done in MATLAB, except mean-value and Shepard interpolation part that was done in C++ and called within MATLAB as needed. To give an idea about

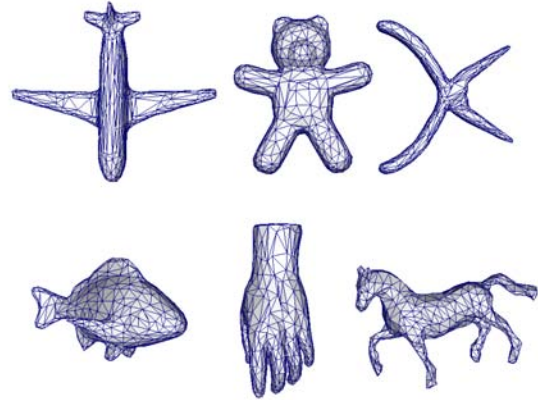


Figure 1: Randomly chosen template surfaces used in the study. These are obtained by simplifying the benchmark models identified by numbers – top row: 306, 13, 359; bottom row: 236, 266, 388.

the timing: evaluating the shape descriptor based on mean-value interpolant takes on average slightly less than 1 minute per template.

We conduct a series of “leave-one-out” experiments: every model in the benchmark is queried against all other models. The ranked result lists generated by the queries are used to compute five retrieval statistics: nearest neighbor (NN), first tier (FT), second tier (ST), discounted cumulative gain (DCG), and average dynamic recall (ADR).

In our first experiment we compare the effects of using projection versus mean-value and Shepard interpolation. Since the mapping domain is the sphere, the descriptor obtained using projection is equivalent to that of [SV01]. Shepard interpolation can be controlled through the use of the parameter α , so we experimented with three choices as shown. The choice was based on the fact that the long distance behavior of mean-value interpolant is similar to that of Shepard interpolant with $\alpha = 2$.

The statistics summarized in Table 1 show that barycentric interpolation and projection show very similar retrieval performances. Surprisingly, on the other hand, Shepard interpolation performs considerably worse than the mean-value interpolation. It is natural to link this to the difference in the order of approximation provided by these interpolation schemes, yet we believe this fact should be investigated further. We also note that the expression for Shepard interpolation has connections to physical force potentials, so the ramifications of our observations to force field based shape descriptors [MDTS08a, GLL07] can be interesting.

In our second experiment we investigate the effect of us-

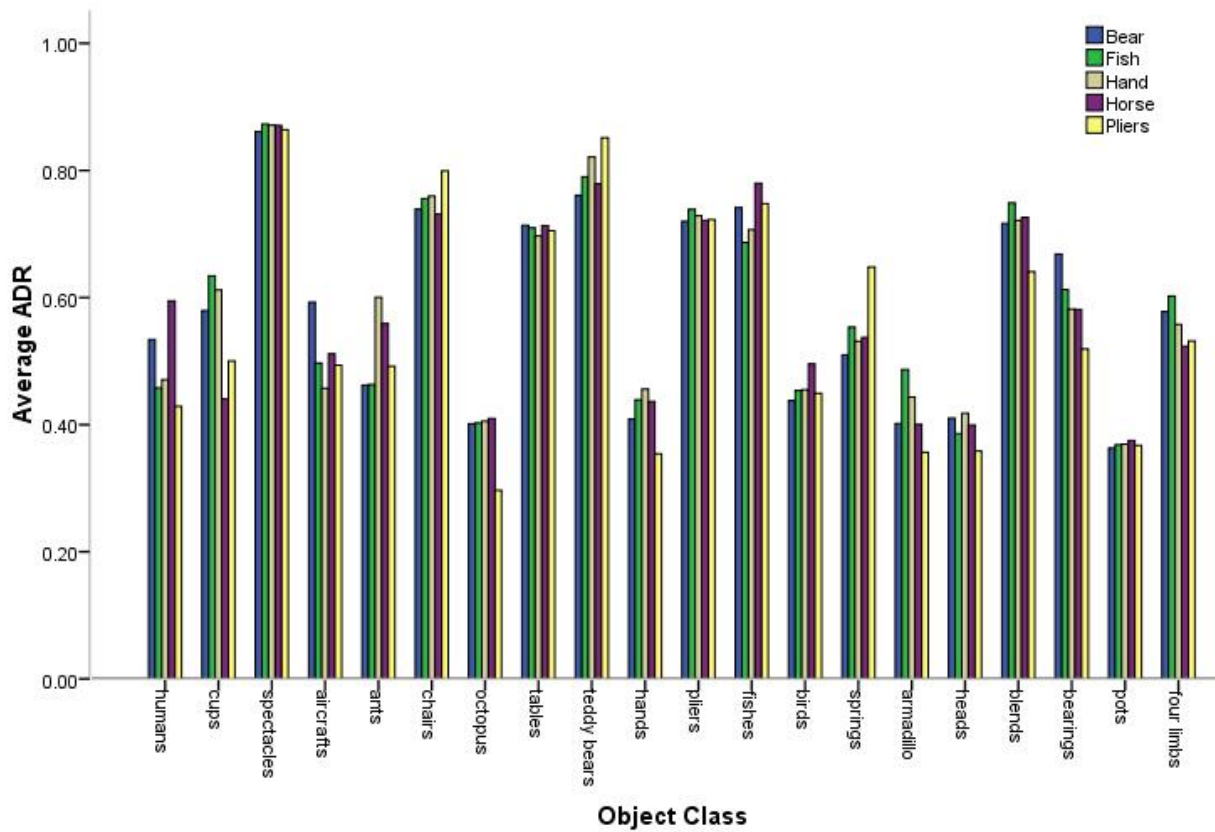


Figure 2: The average ADR values for each class shown by template used.

Mapping	NN	FT	ST	DCG	ADR
Projection	76.8	31.4	45.6	68.3	55.5
Mean-value	73.5	32.3	45.2	68.0	55.7
Shepard $\alpha = 1$	61.0	25.0	39.3	61.5	46.3
Shepard $\alpha = 2$	71.3	26.9	41.8	64.8	50.4
Shepard $\alpha = 3$	72.0	27.3	40.7	65.0	50.8

Table 1: Comparison of various mapping methods. All numbers are percentages.

ing different templates with mean-value interpolation. Table 2 shows the statistics for the templates depicted in Figure 1. The table also shows the retrieval results of combined descriptor obtained by simply concatenating the feature vectors for all the templates. The improved retrieval resulting from such combination shows that the descriptors corresponding to different templates are independent. This point is further corroborated by Figure 2 which shows the average ADR per each class for every template used. Let us note that the tem-

Template	NN	FT	ST	DCG	ADR
Sphere	73.5	32.3	45.2	68.0	55.7
Pliers	72.5	34.5	49.6	68.6	55.6
Horse	78.8	34.8	48.4	70.0	57.9
Teddy-bear	77.0	35.1	49.2	69.7	58.0
Fish	78.8	35.8	49.9	70.5	58.3
Hand	80.5	35.6	49.6	70.9	58.3
All combined	81.3	39.0	55.1	73.3	61.6

Table 2: Using variety of templates for shape retrieval.

plates have also been processed by the normalization applied to the whole benchmark, so they span similar spatial regions. Consequently, the descriptors could have been made much more independent if the templates were differently posed.

4. Summary and Future Work

We have introduced a new shape descriptor which maps the surface features onto an arbitrary template surface using

mean-value interpolation, and extracts a compact numerical feature vector using manifold harmonics of the template. We have shown experimentally that mean-value interpolation is a strong alternative to the prevalently used projection. The usefulness of using different templates was shown, and their independence established.

In the future work we plan to investigate the dependency between the nature of the template and the produced retrieval results. We believe that there does not exist "an ideal" template that would be optimal for all kinds of shape repositories. In fact, the main flexibility offered by our approach is that one can choose optimal templates based on the shape database at hand – protein shapes can benefit from different templates than mechanical engineering parts. Designing approaches for selecting/generating these optimal templates for a given shape database is an interesting direction for future work. We also would like to investigate the possibility of constructing an intrinsically rotationally invariant shape descriptor using similar ideas.

References

- [ASSY07] AKGUL C. B., SANKUR B., SCHMITT F., YEMEZ Y.: Multivariate density-based 3d shape descriptors. In *SMI '07: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2007* (Washington, DC, USA, 2007), IEEE Computer Society, pp. 3–12.
- [BDF*08] BIASOTTI S., DE FLORIANI L., FALCIDIENO B., FROSINI P., GIORGI D., LANDI C., PAPALEO L., SPAGNUOLO M.: Describing shapes by geometrical-topological properties of real functions. *ACM Comput. Surv.* 40, 4 (2008), 1–87.
- [BKS*05] BUSTOS B., KEIM D. A., SAUPE D., SCHRECK T., VRANIĆ D. V.: Feature-based similarity search in 3d object databases. *ACM Comput. Surv.* 37, 4 (2005), 345–387.
- [BNS06] BELKIN M., NIYOGI P., SINDHWANI V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.* 7 (2006), 2399–2434.
- [FKR05] FLOATER M. S., KÓS G., REIMERS M.: Mean value coordinates in 3d. *Comput. Aided Geom. Des.* 22, 7 (2005), 623–631.
- [Flo03] FLOATER M. S.: Mean value coordinates. *Comput. Aided Geom. Des.* 20, 1 (2003), 19–27.
- [GBP07] GIORGI D., BIASOTTI S., PARABOSCHI L.: *Water-tight Models Track*. Tech. Rep. IMATI-CNR-GE 9/07, September 2007.
- [GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (1997), pp. 209–216.
- [GLL07] GENG X., LIU W., LIU H.: Force field based expression for 3d shape retrieval. In *HCI (2)* (2007), vol. 4551 of *Lecture Notes in Computer Science*, pp. 587–596.
- [GW78] GORDON W. J., WIXOM J. A.: Shepard's method of "metric interpolation" to bivariate and multivariate interpolation. *Mathematics of Computation* (Jan 1978).
- [HKS02] HECZKO M., KEIM D., SAUPE D., VRANIĆ D. V.: Methods for similarity search of 3d objects (in german). *Datenbank-Spektrum Zeitschrift für Datenbanktechnologie* 2, 2 (2002), 54–63.
- [IJL*05] IYER N., JAYANTI S., LOU K., KALYANARAMAN Y., RAMANI K.: Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design* 37, 5 (2005), 509–530.
- [JSW05] JU T., SCHAEFER S., WARREN J.: Mean value coordinates for closed triangular meshes. In *TOG(SIGGRAPH)* (2005), pp. 561–566.
- [JZ06] JAIN V., ZHANG H.: Robust 3D shape correspondence in the spectral domain. In *Shape Modeling International* (2006).
- [KFR03] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Symposium on Geometry Processing* (2003).
- [MDSB02] MEYER M., DESBRUN M., SCHRÖDER P., BARR A.: Discrete differential geometry operators for triangulated 2-manifolds. In *Proceedings of Visual Mathematics* (2002).
- [MDTS08a] MADEMLIS A., DARAS P., TZOVARAS D., STRINTZIS M.: 3d object retrieval based on resulting fields. In *Workshop on 3D object retrieval* (April 2008).
- [MDTS08b] MADEMLIS A., DARAS P., TZOVARAS D., STRINTZIS M.: Using ellipsoidal harmonics for 3d shape representation. In *INTERMEDIA Workshop on Hypermedia 3D Internet* (2008).
- [Rus07] RUSTAMOV R.: Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Symposium on Geometry Processing* (2007).
- [RWP06] REUTER M., WOLTER F.-E., PEINECKE N.: Laplace-beltrami spectra as "shape-dna" of surfaces and solids. *Computer-Aided Design* 38, 4 (2006), 342–366.
- [SB07] S. BIASOTTI B., FALCIDIENO P. F. D. G. C. L. S. M. G. P. M. S.: 3d shape description and matching based on properties of real functions. In *Eurographics 2007 Tutorial Notes* (2007), The Eurographics Association, pp. 1025–1074.
- [She68] SHEPARD D.: A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference* (New York, NY, USA, 1968), ACM, pp. 517–524.
- [SV01] SAUPE D., VRANIĆ D. V.: 3d model retrieval with spherical harmonics and moments. In *Proceedings of the 23rd DAGM-Symposium on Pattern Recognition* (London, UK, 2001), Springer-Verlag, pp. 392–397.
- [TV08] TANGELDER J., VELTKAMP R.: A survey of content based 3d shape retrieval methods. *Multimedia Tools and Applications* 39 (2008), 441–471.
- [VL08] VALLET B., LÉVY B.: Manifold harmonics. *Computer Graphics Forum (Proceedings Eurographics)* (2008).
- [VSR01] VRANIĆ D. V., SAUPE D., RICHTER J.: Tools for 3d-object retrieval: Karhunen-loeve transform and spherical harmonics. In *IEEE 2001 Workshop Multimedia Signal Processing* (2001), pp. 293–298.