Vis Comput
DOI 10.1007/s00371-013-0850-0

ORIGINAL ARTICLE

Pose analysis using spectral geometry

Jiaxi Hu · Jing Hua

Abstract We propose a novel method to analyze a set of poses of 3D models that are represented with triangle meshes and unregistered. Different shapes of poses are transformed from the 3D spatial domain to a geometry spectrum domain that is defined by Laplace–Beltrami operator. During this space-spectrum transform, all near-isometric deformations, mesh triangulations and Euclidean transformations are filtered away. The different spatial poses from a 3D model are represented with near-isometric deformations; therefore, they have similar behaviors in the spectral domain. Semantic parts of that model are then determined based on the computed geometric properties of all the mapped vertices in the geometry spectrum domain. Semantic skeleton can be automatically built with joints detected as well. The Laplace–Beltrami operator is proved to be invariant to isometric deformations and Euclidean transformations such as translation and rotation. It also can be invariant to scaling with normalization. The discrete implementation also makes the Laplace–Beltrami operator straightforward to be applied on triangle meshes despite triangulations. Our method turns a rather difficult spatial problem into a spectral problem that is much easier to solve. The applications show that our 3D pose analysis method leads to a registration-free pose analysis and a high-level semantic part understanding of 3D shapes.

Keywords Semantic skeleton extraction · Spectral geometry · Joint detection

J. Hu · J. Hua (✉)
Department of Computer Science, Wayne State University,
5057 Woodward Ave, STE 3010, Detroit, MI 48098, USA
e-mail: jinghua@cs.wayne.edu

1 Introduction

Shape deformation is one of the major research topics in computer graphics. Recently there have been extensive literatures on this topic. There are two major categories of researches on this topic. One is surface-based and the other one is skeleton-driven method. Shape interpolation [2, 9, 11] is a surface-based approach. The basic idea is that given two key frames of a shape, the intermediate deformation can be generated with interpolation and further deformations can be predicted with extrapolation. It can also blend among several shapes. Shape interpolation is convenient for animation generation but is not flexible enough for shape editing due to the lack of local control. The surface-based editing [1, 8] is proposed to bridge this gap. It allows users to control the deformations of a shape with several control points.

Skeleton-driven deformation [3, 13] is another kind of approach. A shape is deformed under the control of an articulated structure, which is more intuitive to human understanding. Skeleton technique is both effective and mature, and widely adopted in industry. He et al. [6] introduced harmonic function to skeleton generation. Users can generate a skeleton of a shape by picking only one or a few reference points manually. De Aguiar et al. [4] constructed skeleton from an animation sequence, in which each frame is represented with a triangle mesh having the same connectivity.

These two categories of methods are not isolated from each other. The surface-based approaches can borrow the skeleton concept as a constraint to guide smooth and realistic deformation [2, 8, 28]; on the other hand, skeleton-driven deformation also uses surface geometry to refine the skeletons [6]. Based on the current state-of-art techniques, there are two major challenges for shape deformation. In most of these techniques, the source data must have the same vertex connectivity. The skeleton generation usually requires users' supervision to achieve a semantic result.

Published online: 01 June 2013

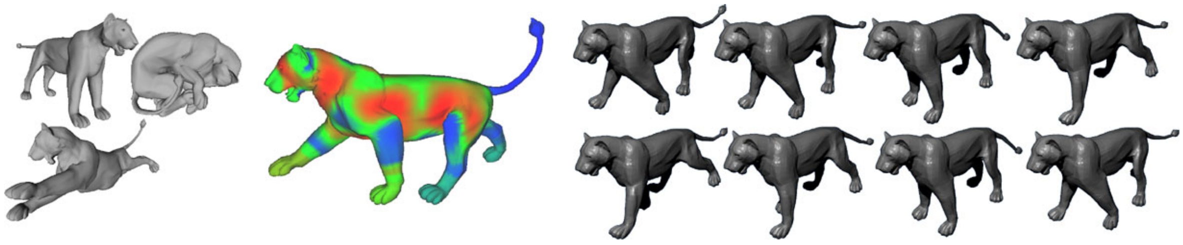


Fig. 1 The procedure of our pose analysis method. Given several unregistered poses of a model which are unregistered and have different triangulations (shown in the *left pane*), a re-embedding from the spatial domain to a geometry spectrum domain is built as shown in the middle. The poses are analyzed in the geometry spectrum domain. The geometric behavior of each point on the pose surface is classified. Then, semantic parts on any poses from the same model can be determined. Colder color in the *middle figure* indicates rigid part on the surface, while warmer color denotes articulated part. With the graph- and skeleton-driven algorithms, the static 3D surface turns into a semantically articulated model which can cast animation

Instead of analyzing shapes in the spatial domain, our method performs a shape pose (motion) analysis in a shape spectral domain induced by Laplace–Beltrami operator on the surface [12, 18, 20]. It is invariant to Euclidean spatial transforms, isometric deformation and different triangulations. It can also be calculated on triangle mesh directly. By calculating the eigenvalues and eigenfunctions of each shape, each point can be transformed into a spectral domain defined by those eigenvalues and eigenfunctions. Thus, our method can study the "pure" geometric information behind those spatial factors. Semantic parts of that model are then determined based on the computed geometric properties of all the mapped vertices in the geometry spectrum domain. Semantic skeleton can be automatically built with joints detected as well. Our method turns a rather difficult spatial problem into a spectral problem that is much easier to solve. Figure 1 briefly illustrates the procedure of our pose analysis method.

Our contributions in this paper are summarized as follows:

- Registration-free pose analysis. We propose a registration-free shape motion analysis method based on Laplace–Beltrami spectral domain. That is to say, the registration between different poses of a model is not required. After the transformation from spatial domain to the spectral domain, all Euclidean transformations, near-isometric deformations, and different mesh triangulations are filtered away, while only "pure" geometric information is left. The pose analysis is conducted in the geometry spectrum domain.
- Semantic surface analysis. Surface mesh vertexes belonging to the same semantic part on different pose surfaces will be mapped to the same coordinates in the geometry spectrum domain, while they carry different spatial properties under different poses. The analysis of the spatial property variation in the geometry spectrum domain will quantify the geometric behaviors of every point during the poses changes, consequently classifying a point to a rigid part or an articulated part in the spectrum domain. The shape is then decomposed into parts with different geometric semantics.
- Automatic semantic skeleton generation with joints identified. The skeleton is generated based on eigenfunctions of the shape. The procedure is automatic without any user interaction. The behavior of the skeleton are constrained by the surface properties and classified surface semantics, which also represents the semantics of that skeleton.

2 Related works

Shape animation and deformation often rely on shape interpolation of two or more key frames of a shape. Certain constrains are considered to make the interpolation as natural as possible while avoiding some artifacts such as local shrinking or collapse. Kilian et al. [11] treated each pose of shapes as a point in a shape space. James and Twigg [9] and Chu and Lee [2] employed mean shift clustering to learn the near-rigid parts of surface from a sequence of poses to guide the interpolation. This kind of methods usually requires one-to-one vertex–face correspondence, either pre-given or obtained by other registration algorithms. The correspondence requirement limits the capabilities of these methods since registration itself is another challenging problem. Skeleton-driven mesh deformation is another popular kind of shape approaches. A shape is deformed under the control of an articulated structure, which is more natural to human understanding [3]. It can provide local control and free deformation. Yan et al. [27] employed simplex transformations to make the skeletons drive the surfaces instead of vertices. Weber et al. [25] used geometric information to guild the skeleton to preserve local details. This kind of approaches usually requires the skeletons to be manually designed to reach a better result. He et al. [6] introduced harmonic function on the surface to build Reeb graph [16, 21] as the skeleton. They reduced the manual operation to picking only one or a few reference points on the surface.

All aforementioned works is from a perspective of spatial analysis. They have to overcome many Euclidean factors such as translation, rotation, and scaling before they analyze the pure geometric properties. Recent research shows that 3D surfaces can also have spectral properties to which the Euclidean factors are not significant. Karni and Gotsman [10] defined mesh Laplacian on polygon meshes based on the adjacent matrix. Reuter et al. [18] introduced Laplace–Beltrami operator to Riemann manifolds represented with surfaces in 3D Euclidean space. The operator is invariant to Euclidean transformations and isometric deformations. The eigenvalues can be used as shape descriptors which is not only invariant but also distinctive. The eigenvalues also contain much information such as the area of the surface, topology, and boundary length. Lévy [12] focused more on the eigenfunctions of the Laplacian equation. The eigenfunctions form a orthogonal basis for the functions defined on the Riemann manifold and can "understand the geometry." A lot of applications can be achieved, such as signal processing on surfaces, geometry processing, pose transfer and parameterization. Rustamov [20] defined a Global Shape Descriptor (GPS) embedding based both on eigenvalues and eigenfunctions and gave a G2 distribution based on the GPS, which can be used as a global shape descriptor stable to topology changes. Hu and Hua [7] analyzed shapes with salient features extracted from the shape spectra.

Our work starts from the perspective of spectral geometry. Therefore, it may extract the pure geometric information behind variant Euclidean factors. The discrete setting makes the Laplace–Beltrami operator can be applied on triangle mesh directly. This saves preprocessing and handles more types of data.

3 Laplacian shape spectrum

In this section, we will briefly review the theory of Laplacian spectrum and describe how to compute it on a triangle mesh. Furthermore, we also discuss the problems based on the spectrum of the real data.

Let $f \in C^2$ be a real function defined on a Riemannian manifold M. Consider the Laplacian eigenvalue equation

$$\Delta f = -\lambda f, \tag{1}$$

where Δ is the Laplace–Beltrami operator and λ is a real scalar. The spectrum is defined to be the eigenvalues arranged increasingly as $0 \le \lambda_0 \le \lambda_1 \le \lambda_2 \le \cdots \le +\infty$. The corresponding eigenvectors offer an orthogonal basis on the manifold. There are more nice properties as follows [18]:

- The operator relies only on the local gradient and divergence on the Riemannian manifold. It is invariant to isometric deformation, Euclidean translation and rotation.
- A scaling factor α applied on the n dimensional manifold will result in that eigenvalues are scaled by $1/\alpha$. The scaling can also be filtered away in the spectrum domain.
- The family of eigenvalues is a descriptor of the manifold. Although there exist two non-isometric shapes which have the same family of eigenvalues, the descriptor can be considered as "unique" in practice.
- The eigenfunctions define an orthogonal basis on the manifold. The larger the eigenvalue is, the higher frequency the corresponding eigenfunction represents. Any function defined on the manifold can be projected to the basis and then transformed into the frequency domain.

Reuter et al. presented a method [18] to solve Eq. (1) with the finite element method. Karni and Gotsman [10] employed graph Laplacian to analyze mesh polygon meshes based on the adjacent matrix. This method relies much on the triangulations. Xu [26], Lévy [12], Rustamov [20] and Vallet and Lévy [24] all used the discrete Laplace–Beltrami operator introduced by Meyer et al. [14]. The operator considers the mesh edge length and triangle surface area as weights, so it is affected much less by the irregular triangulations. Our method also follows this discrete Laplace–Beltrami operator.

The Laplace–Beltrami operator K on discrete triangle meshes is defined as

$$K(p_i) = \frac{1}{2A_i} \sum_{p_j \in N_1(p_i)} (\cot\alpha_{ij} + \cot\beta_{ij})(p_i - p_j), \tag{2}$$

where N_1 denotes one ring neighbor operator; p_i and p_j are vertices with indices i and j; α_{ij} and β_{ij} are the two angles opposite to the edge in the two triangles sharing the edges i, j; A_i is the Voronoi region area of p_i. K is also called mean curvature normal operator because the length of $K(p_i)$ is the mean curvature of p_i and its direction is the direction of the vertex normal of p_i. For all the vertices of a triangle mesh, a Laplace–Beltrami matrix can be constructed as

$$L_{ij} = \begin{cases} -\frac{\cot\alpha_{ij}+\cot\beta_{ij}}{2A_i} & \text{if } i, j \text{ are adjacent,} \\ \sum_k \frac{\cot\alpha_{ik}+\cot\beta_{ik}}{2A_i} & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where α_{ij}, β_{ij}, and A_i are those in Eq. (2) for some i and j. Then, the spectrum problem equation (1) turns into the following eigenvalue problem:

$$L\vec{v} = \lambda\vec{v}, \tag{4}$$

where $\vec{v}$ is an n-dimensional vector. Each entry of $\vec{v}$ represents the function value at one of n vertices on the mesh. It can be regarded as a finite sampling of the eigenfunction at each vertex on the surface. Thus, in this paper, eigenvector and eigenfunction are the same concept. In the following sections, we also call an eigenvector an eigenfunction. Equation (4) is not easy to solve directly because L is not symmetric. Fortunately, the above equation can be represented as a generalized eigenvalue problem which is much easier to solve numerically by constructing a sparse matrix M and a diagonal matrix S such that

$$M_{ij} = \begin{cases} -\frac{\cot\alpha_{ij}+\cot\beta_{ij}}{2} & \text{if } i, j \text{ are adjacent,} \\ \sum_k \frac{\cot\alpha_{ik}+\cot\beta_{ik}}{2} & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases}$$

and $S_{ii} = A_i$. Thus, the Laplace matrix L is decomposed as $L = S^{-1}M$ and the generalized eigenvalue problem is presented as

$$M\vec{v} = \lambda S\vec{v}. \tag{5}$$

As defined above, M is symmetric. S is diagonal and symmetric positive-defined. All the eigenvalues and eigenvectors are real, and the eigenvectors corresponding to different eigenvalues are orthogonal in terms of S dot product:

$$\langle \vec{u}, \vec{w} \rangle_S = \vec{u}^{\mathrm{T}} S \vec{w} \tag{6}$$

where $\vec{u}$ and $\vec{w}$ are eigenvectors of Eq. (5). The orthogonality is represented with

$$\langle \vec{v}_i, \vec{v}_j \rangle_S = 0, \quad i \neq j. \tag{7}$$

Under this setting, the spectrum $\{0, \lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{n-1}\}$ is the family of eigenvalues of the generalized eigenvalue problem defined above. The eigenvectors $\vec{v}_0, \vec{v}_1, \vec{v}_3, \ldots, \vec{v}_{n-1}$ represent the eigenfunctions on the mesh. As it can be seen, the number of eigenvalues and eigenfunctions is reduced from infinite to n, because a triangle mesh is a finite discrete sampling of a continuous surface. It is similar to the discrete Fourier transform and the continuous one. In practice, infinite eigenvalues and eigenfunctions are not necessary. Only first a few of eigenvalues and eigenfunctions are employed to build the geometry spectrum domain.

Figure 2 illuminates the 3rd, 5th and 10th eigenfunctions on different poses. Note that the Laplace–Beltrami operator is defined on continuous manifold, so the triangle meshes are required to be manifolds. They could be either closed manifolds or those with open boundaries, with the same topology. The color turns from cold to warm while the function value grows from a small one to a big one. The eigenfunctions always change along the surface geometry. The three poses are quite different from the spatial view, but the eigenfunctions stay stable on the surfaces. The eigenfunctions rely only on the surface geometry. The shapes are not only different from each other with poses, but also the triangulations. The pose in the first column has about 2000 vertices; the one in the second column has about 10,000; the one in the last column has about 20,000. As it was discussed above, the eigenfunction is also invariant to triangulations. These properties guarantee that the geometry spectrum embedding is invariant to pose deformations and mesh triangulations. In other words, vertices from different poses but at the same position in terms of surface geometry will be embedded together in the geometry spectrum domain, no matter how the poses are deformed or how different the samplings and triangulations are.

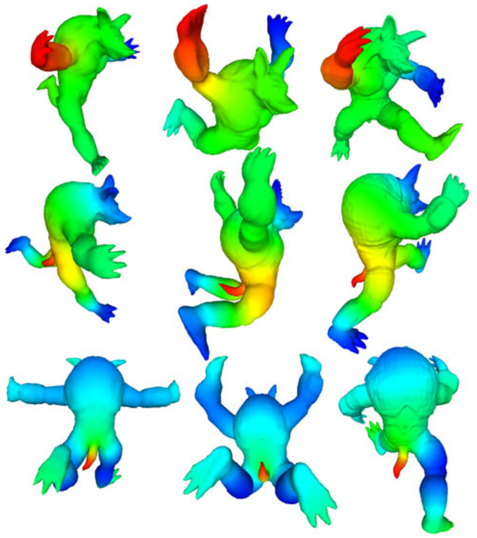


Fig. 2 The 3rd, 5th, and 10th eigenvectors of discrete Laplace matrices in three different poses. Each column demonstrates a pose while each row shows the 3rd, 5th, and 10th eigenvectors from the top to the bottom. The color from *blue* to *green* and then to *red* demonstrates the value changes from small to large. Each eigenvector shows some meaning of the surface. Within a pose, a higher-order eigenvector shows a higher frequency. Note that the pose surfaces in the first column have about 2000 vertices; the ones in the second column have about 10,000 vertices; and the ones in the last column have about 20,000 vertices. The eigenvectors are not only meaningful but also stable to poses and triangulations

The spectrum can describe the intrinsic geometry within the original shape. Theoretically, the shape spectrum is invariant to isometric deformations. However, problems arise when dealing with the real data. Different poses casted by an object are usually near-isometric to each other. The deformations near the joints break the isometric constraint. The computations also bring numerical errors. Dey et al. [5] studied the spectral stabilities under near-isometric deformation. Their results show that the spectra achieved with the cotangent scheme, including the discrete operator in our method, are stable in terms of eigenvalues. Our method produces similar results. Figure 3 lists five shapes represented with triangle meshes while Table 1 lists their first few eigenvalues. The eigenvalues are normalized with the first nonzero eigenvalue to filter away the global scaling according to [18]. Because the first three poses are casted by a same armadillo model, they are considered to be near-isometric to

Table 1 Normalized eigenvalues of different shapes in Fig. 3

Shape	λ_0	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7
Armadillo (a)	0	1	1.23	1.64	2.90	4.37	6.32	8.83
Armadillo (b)	0	1	1.36	1.81	3.20	4.52	6.48	8.51
Armadillo (c)	0	1	1.25	1.33	2.28	4.83	6.76	8.68
Elephant (d)	0	1	2.44	3.07	3.51	3.98	4.24	4.70
Lion (e)	0	1	1.51	2.57	2.66	2.71	4.69	7.92

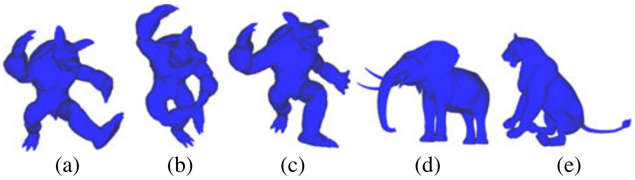


Fig. 3 Five different shapes in the database. The first three shapes are different poses from a same armadillo model. According to Table 1, the three armadillo poses have similar eigenvalues, while the eigenvalues of the elephant and the lion are quite different

each other. This fact is demonstrated by the similar eigenvalues. When the models are different, eigenvalues are dramatically different too, as shown in Table 1. The eigenvalues have enough power to distinguish models and shapes globally. In addition, there are some other potential problems of eigenvectors/eigenfunctions as discussed below which may affect our algorithm. Reuter also discussed these problems in [17]:

- Sign flips occur. If $\vec{v}$ is an eigenvector, $-\vec{v}$ is also one according to the definition. Reuter [17] admitted that sign flips cannot be detected intrinsically on an almost perfect intrinsic symmetric shape. We employ the absolute value to avoid the sign flip problem.
- Eigenvectors switch. The neighbor eigenvalues may switch due to the perturbations of the deformations and numerical computations. So are the corresponding eigenvectors. It happens nearly on every mesh. Reuter [17] gave a solution to reorder the eigenvectors based on the Morse–Smale graph. We use the same scheme. Without further notation, all the eigenvectors, in the rest of the paper, refer to the reordered ones.
- Higher dimensional eigenspaces can theoretically occur. However, they rarely happen in practical data. We have not found any example in our results so far.
- Duplicated eigenvalues may exist. A highly symmetric shape, e.g. sphere or cube, has duplicated eigenvalues. The linear combinations of the corresponding eigenvectors are also eigenvectors. Nevertheless, practically used animation models do not have such high symmetry. That is to say that duplicated eigenvalues rarely happen practically in our application.
- Low frequency eigenvectors are stable under near-isometric deformation. Reuter [17] had a detailed discussion about the stabilities of the shape spectrum with respect to near-isometric deformations and noises, and used direct spectral embedding for the semantic shape segmentation. Our experiments also show that the low-frequency eigenvectors are quite stable. The third row of Fig. 2 demonstrates the stability of the 10th eigenvector for different poses. Although the spectra are stable globally, the local values of eigenvector may shift. This usually happens when there is a twisting deformation. The shifting will affect the registration accuracy under this single frequency. However, with multi-frequency embedding and multiple shape data, the accuracy will be corrected by other values that are stable.

The above discussion shows the near-isometric shapes will have similar behavior in the spectral spaces.

4 Geometry spectrum domain embedding

The Laplace–Beltrami operator defines a family of eigenvalues and a family of eigenfunctions. The eigenvalues can be used as shape descriptors which are stable and distinctive. It also contains “frequency” information. The smaller eigenvalues denote lower frequencies. The eigenvectors form an orthogonal basis on the manifold. Any functions can be projected to the basis and reconstructed with the linear combination of these eigenfunctions. All of these are global analysis in the spectrum domain. However, the main goal here is to study the local behaviors of the surfaces. It is obvious that each eigenfunction ϕ_k is assigned a real value at every surface point p as $\phi_k(p)$. With respect to each point, there exists a mapping from a point on the surface in 3D spatial space to an infinite geometry spectrum space as:

$$G_S(p) = \left(\frac{\phi_1(p)}{\sqrt{\lambda_1}}, \frac{\phi_2(p)}{\sqrt{\lambda_2}}, \frac{\phi_3(p)}{\sqrt{\lambda_3}}, \dots\right), \tag{8}$$

where p is a point on a surface S and ϕ_k is the kth eigenfunction corresponding to the kth eigenvalue λ_k of S. Each eigenvector is normalized by

$$\langle \vec{v}_i, \vec{v}_j \rangle_S = 1, \quad i = 1, 2, 3, \dots, \tag{9}$$

on surface S. As we work on the poses casted by the same object, the scales of each surface can be normalized. Thus, the scales of the values in eigenfunctions represent the geometries of the shapes. We summarize some major advantages of this geometry spectrum domain embedding as follows:

- If the surface in 3D space has no self-intersections, the embedding has no self-intersections in the infinite domain either. That means $G_S(p_i) = G_S(p_j)$ if and only if $p_i = p_j$ on S.
- The embedding is based only on eigenfunctions on the manifold. It relies only on the manifold metric and is invariant to the Euclidean embedding in 3D space of S. The embedding mapping filters away the Euclidean transformations and near-isometric deformations.
- The embedding is invariant to different triangulations because of the implementations of discrete Laplacian.

With this embedding, our method does not require any preprocessing such as normalization, remeshing or registration. All these spatial factors do not mater in the geometry spectrum domain. With the same surface, there is only one basis set for the embedding. Nevertheless, it may not be true for different poses. The eigenfunctions of the manifold satisfies Eq. (1). If a certain function ϕ_k is a normalized eigenfunction corresponding to some eigenvalue λ_k, then, according to this equation, $-\phi_k$ is also a normalized eigenfunction. The experiments also show that eigenfunctions from different poses can flip with sign corresponding to the same eigenvalue. In order to overcome this flipping problem, the mapping is restricted as an absolute one as:

$$AG_S(p) = \left(\left\| \frac{\phi_1(p)}{\sqrt{\lambda_1}} \right\|, \left\| \frac{\phi_2(p)}{\sqrt{\lambda_2}} \right\|, \left\| \frac{\phi_3(p)}{\sqrt{\lambda_3}} \right\|, \dots \right). \tag{10}$$

The absolute mapping will break the first property about self-intersection (see above), and the symmetric points on the surface will be mapped together. In our framework, it is natural to assume those parts have similar physical behaviors when they are symmetric on the surface. Thus, the absolute mapping does not affect the accuracy of mapping in terms of symmetry. For example, the left and right elbows have the symmetric geometry properties, the registration across each is acceptable, as we do not require the dense mapping and registration. In theory, the embedding is a high-dimensional space; however, in practice we choose first a few eigenvectors. Rustamov [20] suggested first d dimensions. Reuter [17] used first 3 to 6 eigenfunctions. Our experiments show that the low-frequency eigenvectors are globally stable, which is shown in Fig. 2. According to the computation, high-frequency eigenvectors become relatively unstable due to the near-isometric deformations and numerical errors. In our experiments, we use first 3 to 10 eigenvectors for the spectral embedding.

5 Semantic shape analysis

5.1 Semantic point classification

The geometry spectrum embedding transforms each point on the surface from the Euclidean space to an infinite geometry spectrum space. Suppose there exists a spatial surface which is near-isometrically deformed along time, denoted by $S(t)$. The points with the same positions relative to the surfaces S at different times will map into the same coordinates in the spectrum space, despite the different locations, orientations and poses of the original surfaces. Although the point is fixed in the spectral domain, it can carry varying geometric properties on S. That is to say, in the spectral domain, the properties at a point vary while the pose changes. For each point p in the spectral domain, we can define a property function $f_p(t)$ which depends only on time t. Imagine that if the properties are chosen to be invariant to Euclidean transformations but only sensitive to pose changes, what can be observed in the spectrum domain is that properties vary on certain regions along with pose changes while not on the other regions. The former situation indicates articulations while the latter one indicates rigid parts of the original shape. There are some well studied features on surfaces, such as curvatures, normals, geodesic fans, and so on. In our framework, mean curvatures are a straightforward choice, as the Laplacian operator is also the mean curvature normal operator on the surface. When the shape deformation $S(t)$ is given, the pose behaviors of all the points can be classified into the articulate or rigid.

The data in our framework is not a continuous surface changing with time but N frames of meshes. N could range from 2 to 10 or even more. The property functions are reduced to a discrete set. A triangle mesh is a discrete sampling of a surface, therefore, an exact correspondence of a vertex may not exist on another near-isometric mesh. Thus, the property set on a vertex is built based on an approximation. Suppose f_p is a feature set which is going to be built at a vertex p on the surface S. First, the embedding of p is calculated, the mean curvature of p is put into f_p as an element. Then, for each following frame of meshes S_i, a point p_i is found as a nearest one to p in the geometry spectrum domain base on the Euclidean distance, and the mean curvature of p_i is put into f_p as another element. Therefore, the element in f_p can classify the geometric behaviors of p through different poses. Figure 4 illustrates the maximum, minimum and range distributions on the surface among different poses.

5.2 Property smoothing

After Eq. (5) is solved, vertices can be mapped from spatial space into to geometry spectrum domain directly with the indices. The chosen properties can be assigned in the spectrum domain. As mentioned above, mean curvature is chosen because the mean curvature normal vector can be obtained by multiplying the Laplacian matrix with the vertex position matrix. Mean curvature is also invariant to Euclidean transformation. However, directly assigning the mean curvature will cause stability problems. The embedding is applied on each discrete vertex. A particular vertex usually cannot find the exact matching with other vertices from other surfaces but has to use the neighbor information in the geometry spectrum domain. Based on the definition, mean curvatures obtained by the mean curvature normal operator use only one ring neighborhood on the mesh. When the mesh is constructed, noise could be involved during the modeling or reconstruction procedures. Thus, the direct mean curvatures will contain a lot of local variance, which will affect the accuracy and stability of the pose analysis in the spectrum domain. Therefore, they have to be smoothed first.

The smoothing process is done with Laplacian eigenfunctions. As it is discussed in the Laplace spectrum section, any function f defined on the surface can be transformed into frequency domain by projecting it onto the eigenfunctions. The coefficient family $\{c_i\}$ forms the frequency spectrum of f as its counterpart in 1D, which is well known as Fourier transform. The smoothing is done by applying a low-pass filler in the frequency domain then transforming filtered coefficients back to the surface function. Figure 5 illustrates the mean curvature reconstruction procedure with different numbers of eigenfunctions. As is shown, the reconstruction with first 130 eigenfunctions is usually sufficient.

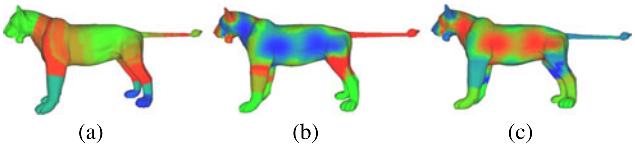


Fig. 4 Mean curvatures values in spectral pose analysis: (**a**) maximum mean curvature distribution on each vertex during pose transformation; (**b**) minimum mean curvature distribution; (**c**) mean curvature range distribution. The values are histograms equalized for visualization

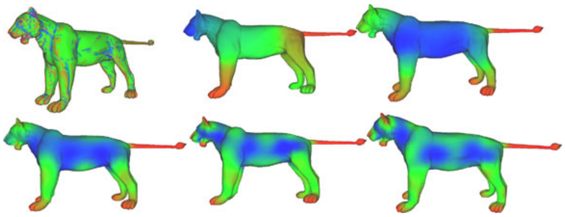


Fig. 5 Mean curvature reconstruction on eigenfunctions of the manifold. From left to right, top to bottom, the *first figure* shows the discrete mean curvatures on the surface, obtained by applying Laplace—Beltrami operator on the Euclidean embeddings; the rest of figures are reconstructions with the first 6, 20, 50, 100, and 130 eigenfunctions

6 Automatic skeleton and joint extraction

When all the points on the surface shape are classified and clustered into semantic parts, it allows an automatic skeleton construction with joint identification. Here we adopt Reeb graph to achieve this goal.

6.1 Skeleton extraction

Reuter [17, 19] discussed the skeletal representation based on the eigenfunctions. We found that in the practical data, the intersections of different parts are not stable if the centers of Reeb graph are employed directly. They may shift away from the semantic locations where they should be. The experiments show that iteratively shrinking the mesh to the center produces smoother results. Our skeleton construction is automatic with two simple steps as demonstrated in Fig. 6.

Iso-contour shrinking. For each vertex on the mesh, the contour with the same function value of the vertex is traversed and found. Then, the vertex is moved to the geometric center of the isocontour. This results in a skeleton-like mesh.

Skeleton construction. Applying the algorithm in [15] on the shrunk mesh with the original eigenfunction, because the mesh is shrunk to the skeleton shape, the spatial embedding of the Reeb graph is accurate enough to become a skeleton.

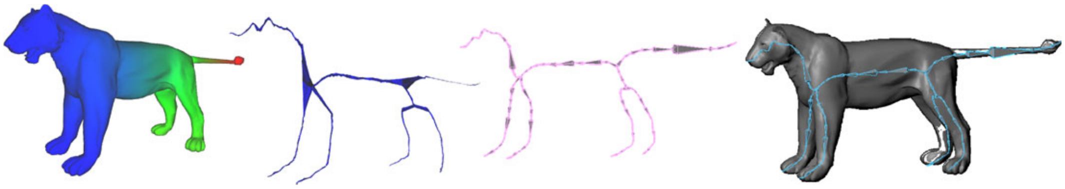


Fig. 6 Automatic skeleton generation. From left to right: the first non-trivial eigenfunction of the loin model; shrink mesh based on isocontours; skeleton generated with Reeb graph algorithm on the shrunk shape; and the embedding of the skeleton within the original model

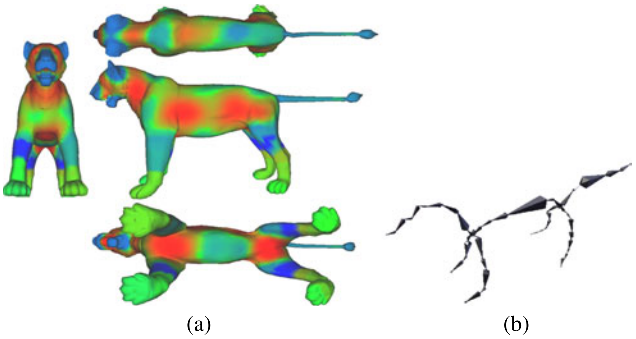


Fig. 7 Mean curvature range distributions on a lion model and the extracted semantic skeleton with joints identified based on the distribution

6.2 Joint detection

Based on the changing geometric behaviors of points in the geometry spectrum domain, we are able to automatically spot the joints as long as the deformations around the joints are presented in the given deformation sequence. Figure 4 demonstrates the basic ideal of the pose analysis in the geometry spectrum domain. Figure 4(a) is the distribution of the mean curvature maxima on the surface. The larger the value is, the more the surface on that point can bend along relative to the negative direction of the normal at that point. Figure 4(b) is the distribution of mean curvature minima. It predicts the behavior that surface bends along the positive normal direction. Note that the values on the surface are histogram equalized. The same color does not mean the same value across different surfaces. Ideally, if a part is always rigid during pose transformation, the geometric shape will never change. A point on that part has the exact constant mean curvature all the time. Thus, the minimum and maximum of mean curvature are equal to each other. On the contrary, if a part varies, the minima and maxima will fall away from each other. This mean curvature change range is a measurement describing how "rigid" the point and its neighborhood is, which is shown in Fig. 4(c). The result is very natural. The articulations like neck have different forms under different poses. The parts like nose will not change too much during different poses. Figure 7 shows the complete example.

7 Experiments and applications

In this section, we show some experiment results of skeleton and joint extraction as well as some further applications based on the semantic skeletons. Note that the pose shapes are represented with triangle meshes. In our experiments, we use mesh data sets from SHREC07 and the one Sumner and Popović [22] used.

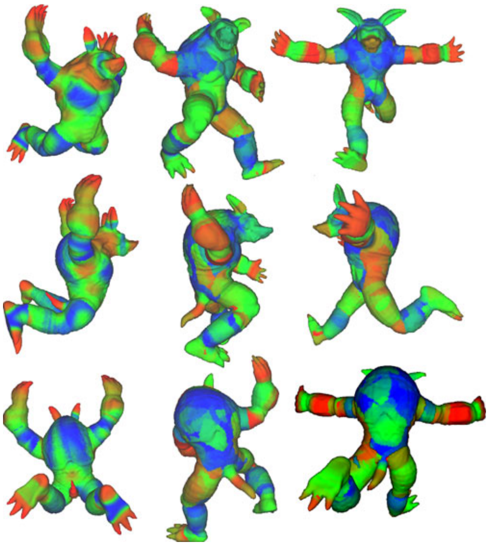


Fig. 8 Mean curvature range distributions on armadillo models. The chest and back shell usually stay rigid while the neck, elbows, and waist vary during pose changes.

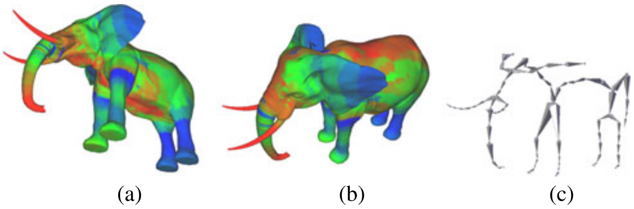


Fig. 9 Mean curvature range distributions on an elephant model and the extracted semantic skeleton with joints identified based on the distribution

7.1 Skeleton and joint extraction

Figure 8 gives an example of the armadillo shape. The main body, especially the chest and the back shell, will not have much variance when it casts different poses. Instead, when the armadillo often changes its postures of head, arms or legs, the neck, shoulder, and waist follow the pose changes. The mean curvature ranges on the surface lead to a segmentation directly, which segments the rigid parts and articulations apart. With the help of the mean curvature ranges, hierarchy graphs can be built as is described in [2]. Figure 9 shows another example.

7.2 Animation

Skeleton-driven deformation has been extensively studied. It is intuitive to human understanding. Most of poses of creatures are controlled by bones and muscles and then represented by the skin surfaces. The technique of the skeleton-driven deformation and animation is widely used in the animation and gaming industry. The classical pipeline is as following: first, manually design a skeleton of a mesh surface;

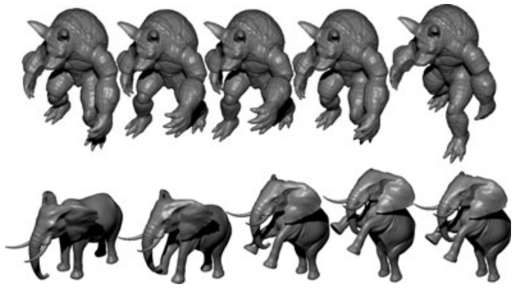


Fig. 10 Animation sequence. With the automatically extracted semantic skeleton, user can edit the pose freely. The animation sequence can be generated among the edited key frames

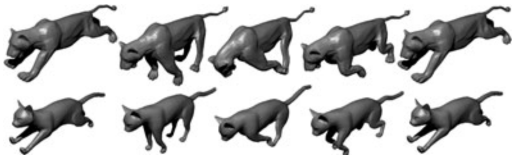


Fig. 11 Motion transformation from a lion model to a cat model

then, assign the vertices of that surface to semantic skeletal parts; then, deform the mesh along the skeleton. Fortunately, our method automatically classifies semantic parts of surfaces during pose changes, and then produces graphs that can be treated as skeletons of meshes. The vertices of the semantically classified surface are automatically associated with skeletal parts with joints identified. There are many existing algorithms that can be employed to deform and control such a shape with skeletons. Figure 1 has already given an example. The skeletons are learnt from several key frames, but can control the shape to cast much more poses than that. Figure 10 also shows some other deformation sequences. These new poses are not just any ones in the reference frames, but some potential possibilities the models can cast based on the knowledge from existing frames.

7.3 Pose transformation

Pose transformation is another popular graphics application. The motivation is obvious. If a pose can be transformed from one shape to another similar shape automatically, a lot of time can be saved by modeling one key shape instead of modeling lots of different shapes. In our framework, the poses are represented with semantic parts. Two similar shapes will have similar semantic parts and skeletons. Graph or skeleton matching algorithm, such as in [23] can find the correspondence between two similar skeletons. After that, a pose driven by a skeleton can be transformed to a similar pose with a corresponding skeleton. Figure 11 demonstrates how running poses are transformed from a lion model to a cat model.

8 Conclusion

Differently from the existing spatial approaches, our method allows to understand the poses in the geometry spectrum domain. The geometry spectrum is based on the eigenvalues and eigenfunctions that are defined by the Laplace–Beltrami operator on the surface. The Laplace–Beltrami operator relies only on the metric on the surface, therefore it is invariant to Euclidean translation, rotation and scaling. It is also invariant to isometric deformations. Thus, the eigenvalues, eigenfunctions, and the geometry spectrum domain share the invariance. Ideally, every point on a spatial surface should be embedded in the geometry spectrum domain only by its geometric meaning. As long as the poses casted by one model are near-isometric to each other, they will be re-embedded to a uniform surface in the infinite geometry spectrum. In practice, the shape spectrum is stable under the near-isometric deformations. For example, the points on the elbow of the model will always be embedded around a common location in the spectrum domain, no matter how the model's pose changes. The spectrum reflects the intrinsic characteristics of a surface despite varying Euclidean space embeddings.

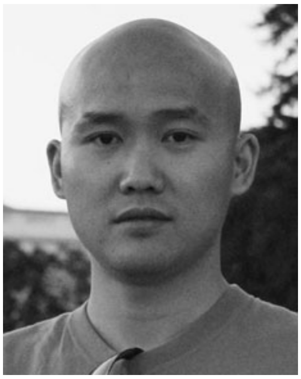
The discrete setting makes it possible and easy to apply the Laplace–Beltrami operator directly on the surfaces represented by triangle meshes. The continuous Laplacian equation turns into a symmetric generalized sparse matrix eigenproblem. The eigenvalues are kept the same within a finite number, and eigenfunctions are represented with eigenvectors as area-weighted samplings. This also makes the spectrum domain invariant to different sampling rates and triangulations.

Our method analyzes data without preprocesses like remeshing or registration. It first transforms spatial surfaces into geometry spectrum domain. Each point is mapped along with its spatial geometry properties. The properties are smoothed with a low-pass filter defined on the basis of eigenfunctions. In the spectrum domain, each point carries a set of properties during the pose variations. It is efficient to classify points on the surface into rigid parts and articulated parts by analyzing the geometric property changes on those points mapped in the geometry spectrum domain. The eigenfunction can also provide rich geometric meaning, which leads to an automatic semantic skeleton with joints identified. The experimental results show that the filtered mean curvature range can predict different semantics of parts on the original surface. It may be very useful in motion analysis in computer vision and pattern-recognition tasks as well.

References

1. Au, O.K.-C., Fu, H., Tai, C.-L., Cohen-Or, D.: Handle-aware isolines for scalable shape editing. ACM Trans. Graph. **26**(3), 83 (2007)

2. Chu, H.-K., Lee, T.-Y.: Multi-resolution mean shift clustering algorithm for shape interpolation. IEEE Trans. Vis. Comput. Graph. (2009)
3. Cornea, N.D., Silver, D., Min, P.: Curve-skeleton properties, applications, and algorithms. IEEE Trans. Vis. Comput. Graph. **13**(3), 530–548 (2007)
4. de Aguiar, E., Theobalt, C., Thrun, S., Seidel, H.-P.: Automatic conversion of mesh animations into skeleton-based animations. Comput. Graph. Forum **27**(2), 389–397 (2008)
5. Dey, T.K., Ranjan, P., Wang, Y.: Convergence, stability, and discrete approximation of Laplace spectra. In: Proceedings of the Twenty-First Annual ACM–SIAM Symposium on Discrete Algorithms, SODA '10, pp. 650–663. Society for Industrial and Applied Mathematics, Philadelphia (2010)
6. He, Y., Xiao, X., Seah, H.-S.: Harmonic 1-form based skeleton extraction from examples. Graph. Models **71**(2), 49–62 (2009)
7. Hu, J., Hua, J.: Salient spectral geometric features for shape matching and retrieval. Vis. Comput. **25**(5–7), 667–675 (2009)
8. Huang, J., Shi, X., Liu, X., Zhou, K., Wei, L.-Y., Teng, S.-H., Bao, H., Guo, B., Shum, H.-Y.: Subspace gradient domain mesh deformation. ACM Trans. Graph. **25**(3), 1126–1134 (2006)
9. James, D.L., Twigg, C.D.: Skinning mesh animations. ACM Trans. Graph. **24**(3) (2005)
10. Karni, Z., Gotsman, C.: Spectral compression of mesh geometry. In: International Conference on Computer Graphics and Interactive Techniques, pp. 279–286 (2000)
11. Kilian, M., Mitra, N.J., Pottmann, H.: Geometric modeling in shape space. ACM Trans. Graph. **26**(3), 64 (2007)
12. Lévy, B.: Laplace–Beltrami eigenfunctions: towards an algorithm that understand s geometry. In: IEEE International Conference on Shape Modeling and Applications (2006). Invited Talk
13. Lewis, J.P., Cordner, M., Fong, N.: Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In: SIGGRAPH '00: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, pp. 165–172 (2000)
14. Meyer, M., Desbrun, M., Schröder, P., Barr, A.: Discrete differential geometry operators for triangulated 2-manifolds. In: VisMath (2002)
15. Pascucci, V., Scorzelli, G., Bremer, P.-T., Mascarenhas, A.: Robust on-line computation of Reeb graphs: simplicity and speed. ACM Trans. Graph. **26**(3), 58 (2007)
16. Patanè, G., Spagnuolo, M., Falcidieno, B.: A minimal contouring approach to the computation of the Reeb graph. IEEE Trans. Vis. Comput. Graph. **15**(4), 583–595 (2009)
17. Reuter, M.: Hierarchical shape segmentation and registration via topological features of Laplace–Beltrami eigenfunctions. Int. J. Comput. Vis. **89**(2), 287–308 (2010)
18. Reuter, M., Wolter, F.E., Peinecke, N.: Laplace–Beltrami spectra as"Shape-DNA" of surfaces and solids. Comput. Aided Des. **38**(4), 342–366 (2006)
19. Reuter, M., Wolter, F.-E., Shenton, M., Niethammer, M.: Laplace–Beltrami eigenvalues and topological features of eigenfunctions for statistical shape analysis. Comput. Aided Des. **41**(10), 739–755 (2009)
20. Rustamov, R.M.: Laplace–Beltrami eigenfunctions for deformation invariant shape representation. In: SGP '07: Proceedings of the Fifth Eurographics Symposium on Geometry Processing, pp. 225–233 (2007)
21. Shinagawa, Y., Kunii, T.L.: Constructing a Reeb graph automatically from cross sections. IEEE Comput. Graph. Appl. **11**(6), 44–51 (1991)
22. Sumner, R.W., Popović, J.: Deformation transfer for triangle meshes. In: SIGGRAPH '04: ACM SIGGRAPH 2004 Papers, pp. 399–405 (2004)
23. Sundar, H., Silver, D., Gagvani, N., Dickinson, S.: Skeleton based shape matching and retrieval. In: Shape Modeling and Applications, pp. 130–139 (2003)
24. Vallet, B., Lévy, B.: Spectral geometry processing with manifold harmonics. Comput. Graph. Forum (2008)
25. Weber, O., Sorkine, O., Lipman, Y., Gotsman, C.: Context-aware skeletal shape deformation. Comput. Graph. Forum **26**(3) (2007)
26. Xu, G.: Discrete Laplace–Beltrami operator on sphere and optimal spherical triangulations. Int. J. Comput. Geom. Appl. **16**(1), 75–93 (2006)
27. Yan, H.-B., Hu, S., Martin, R.R., Yang, Y.-L.: Shape deformation using a skeleton to drive simplex transformations. IEEE Trans. Vis. Comput. Graph. **14**(3), 693–706 (2008)
28. Zou, G., Hua, J., Muzik, O.: Non-rigid surface registration using spherical thin-plate splines. In: Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 367–374 (2007)



Jiaxi Hu is a Ph.D. student at Department of Computer Science at Wayne State University. He is currently a member of the Graphics and Imaging Laboratory. He received his M.S. degree and B.S. degree from Huazhong University of Science and Technology at Wuhan, China in 2005 and 2003, respectively. His current research interests include geometric computing, computer graphics and scientific visualization.

Jing Hua is an Associate Professor of Computer Science at Wayne State University and the founding director of Graphics and Imaging Lab (GIL). Dr. Hua received his Ph.D. degree (2004) in Computer Science from SUNY at Stony Brook. His research interests include geometric computing, computer graphics, visualization, 3D image analysis and informatics. He received the Gaheon Award for the Best Paper in International Journal of CAD/CAM in 2009, the Best Paper Award at ACM Symposium on Solid Modeling and Applications 2004, and the K.C. Wong Research Award in 2010. He is a Principal Investigator for research projects funded by the National Science Foundation, National Institutes of Health, State of Michigan, and Ford Motor Company.