The Visual Computer manuscript No.
(will be inserted by the editor)

Ariel Shamir · Lior Shapira · Daniel Cohen-Or

Mesh Analysis using Geodesic Mean-Shift

the date of receipt and acceptance should be inserted later

Abstract In this paper we introduce a versatile and robust method for analyzing the feature space associated with a given mesh surface. The method is based on the mean-shift operator which was shown to be successful in image and video processing. Its strength lays in the fact that it works in a single joint space of geometry and attributes called the *feature-space*. The mean-shift procedure works as a gradient ascend finding maxima of an estimated probability density function in feature-space. Our method for using the mean-shift technique on surfaces solves several difficulties. First, meshes as opposed to images do not present a regular and uniform sampling of domain. Second, on surfaces meshes the shifting procedure must be constrained to stay on the surface and preserve geodesic distances. We define a special local geodesic parameterizations scheme, and use it to generalize the mean-shift procedure to unstructured surface meshes. Our method can support piecewise linear attribute definitions as well as piecewise constant attributes.

Keywords mean-shift, meshes, segmentation, feature extraction

1 Introduction

Data partitioning and feature identification have become a fundamental part of many applications in graphics and visualization. In this paper we introduce a versatile and robust method that allows filtering attributes, extracting features and partitioning a given surface. The method is based on the analysis of a *feature-space* using the *mean-shift* procedure. The feature-space is defined as a high dimensional space associated with the surface geometry and its attributes. The coordinates of a point in this space include a combination of the *spatial* coordinates and a set of associated *attributes* that are considered in the analysis. The mean-shift is defined as a gradient ascend search for maxima of a density function

Supported by the Israeli Ministry of Science and Education Grant No. 2003088

Ariel Shamir
Efi Arazi School of Computer Science
The interdisciplinary center
E-mail: arik@idc.ac.il

Lior Shapira
School of Computer Science
Tel-Aviv University
E-mail: lior@liors.net

Daniel Cohen-Or
School of Computer Science
Tel-Aviv University
E-mail: dcor@tau.ac.il

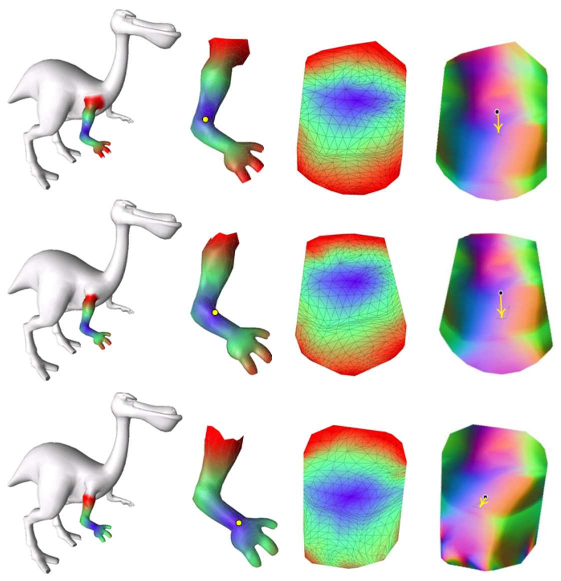
Fig. 1 An overview of the geodesic mean-shift procedure on surface meshes. The local geodesic neighborhood around each vertex is flattened. The area is shown from blue (center) to red (periphery) on the mesh, and after parametrization. Using the parameterized mesh vertices, the attributes are mapped to color channels and the mesh is rasterized to create the feature-space neighborhood. In this example we mapped the XYZ of the normals as a 3D feature vector to RGB. The new mean is computed by excluding values outside the given window range in both spatial and attribute sub-spaces. The window is shifted to the new mean and the process continues until convergence (from top to bottom in this figure).

over the feature-space. It operates by iteratively shifting a fixed size window to the average of the data points within it, until convergence. The end result of the procedure is a mapping from each point in feature-space to *modes* which are, loosely speaking, the most likely values in their neighborhood. This mapping can then be utilized for clustering, filtering or feature identification.

The strength of the mean-shift approach lies in the fact that it simultaneously considers the combined feature space composed of both geometry and attributes together. Unlike many clustering techniques, it does not require an initial guess for the location or the number of the modes or clusters. Instead, the modes are determined by the mean-shift procedure itself. The mean-shift has been shown to be successful in image and video processing [5,7,40], and for unstructured volumetric meshes using a continuous piecewise constant feature-space [32].

In this paper we define a novel scheme of the mean-shift procedure for surface meshes, which we term *Geodesic Mean-Shift*. We use the most pervasive representation in graphics which is a triangular mesh, and apply the mean-shift operator to the mesh vertices rather than to the mesh faces. Several difficulties arise when moving to this new geometric domain. First, meshes as opposed to images do not present a regular and uniform sampling of the domain. Second, on surface meshes simple averaging of 3D coordinates may cause the result to be outside the mesh. This means that the shifting procedure must be constrained to the mesh domain, and the averaging must use correct distance calculations between points, that is, *geodesic* distances instead of Euclidean. The third difficulty is a result of applying the mean-shift to vertices rather than faces of the mesh. This imposes a piecewise linear feature space rather than piecewise constant as in pixels or volumetric elements in previous works.

The fundamental operation in the mean-shift procedure is the computation of the weighted mean of the points in a specific neighborhood around a center point. When we move to meshes and use continuous feature-space instead of a discrete one, discrete averaging based on counting samples in the neighborhood cannot be used anymore. Remembering that the discrete averaging originated from a Parzen window estimation method for the density function [27], we replace the averaging with a continuous estimation based on integration. Hence, in our continuous space we consider the whole volume inside the Parzen window instead of only counting discrete samples. When the function is piecewise constant, the averaging uses the area (or volume) as a weighting factor for each constant value. When the function is piecewise linear, we sample the whole window area (or volume) uniformly by employing rasterization with linear blending.

To constrain the movement of the mean-shift on a surface mesh, we use a new type of local parametrization. Unlike common parametrization techniques [13], which goal is to cover large areas with global constraints, our *Geodesic parametrization* is local and emphasizes the center point. Its goal is to create a flattened neighborhood around each of the mesh vertices which preserves the geodesic distances to the center vertex as much as possible, and distorts the area around the vertex as little as possible. These parametric maps are then used in the mean calculation. The distances from the window center point on the 2D map are used as an approximation to the geodesic distances on the 3D surface. To achieve a fast approximation of the piecewise linear attribute neighborhood, the map is rasterized to create a sampling around the vertex (see Figure 1).

Our primary contributions are as follows:

- We generalize the mean-shift procedure to be applicable to a mesh-surface embedded in 3D, extending its applicability from regular image-space to unstructured and irregular meshes.
- We introduce a moving geodesic parametrization, enabling the mean-shift to be performed as if the surface was locally flat.
- We extend the mean-shift procedure to work on a continuous feature-space which is piecewise linear and not only piecewise constant using area sampling.

2 Previous Work

Clustering primitive elements into large regions of similar attributes has been used for surface analysis. In particular segmentation, partitioning and feature extraction are used as a first stage in applications such as parametrization, simplification, compression, matching, morphing and more [18, 37,25,26,28,20,30,3,31,23]. Often clustering is performed in a greedy manner by assigning a triangle to the closest seed [20], by adding triangles incrementally to regions [26, 19,38], or by merging whole regions at a time [15,8,10]. Mean-shift provides a more robust approach to the analysis of meshes since it takes a global view based on the search for most likely values in various regions of the mesh in the joint geometric and attribute space.

Partitioning, clustering and segmentation for the analysis of data appear in many fields such as image processing, vision, data-mining and machine-learning [1]. Clustering approaches are classified to ‘parametric’ and ‘non-parametric’ [29]. In the parametric approach, the number of clusters (i.e., the parameter) is either known or preset in advance, and all elements (or points) are assigned to one of them. Examples of such methods are the K-means procedure (see for example [17]) and the Gaussian mixture model [42]. Non-parametric approaches do not rely or preset the number of clusters, hence no parameter is needed. The number of clusters can change as the algorithm progresses, becoming one of the results of the algorithm along with the partitioning itself. Examples of algorithms which use this approach are K-nearest neighbors, bilateral filtering [39] and the mean-shift [5,16]. The main difference between the two is that mean-shift employs a moving window while in bilateral filtering the window is fixed. Furtheremore, mean-shift does not rely on outside stopping criteria, but rather converges to the local density maxima.

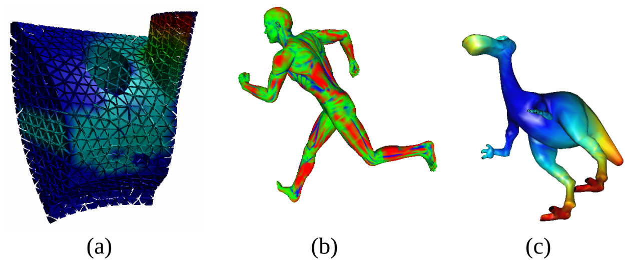

Fig. 2 Attributes on meshes can be predefined and given such as temperature on a tetrahedral mesh (a), or extracted such as mean curvature (b) or centricity (c) on a surface mesh. Centricity is calculated as average geodesic distance from each vertex to all other vertices.

Parametrization is an active field of research in Graphics. In several important papers Floater presented barycentric coordinates as a way to allow vertices in triangulations to be expressed as convex combinations of their neighbors [11]. This was later enhanced by the the use of mean value coordinates in [12]. Such constraints can be solved as a sparse linear system of equations. In [24], Levy and Mallet presented a method to solve the same convex combination problem as a minimization problem. Several papers such as Sheffer and Sturler [36], and Zigelman et al. [43], concentrate on parametrization with free boundaries. Other works have focused on processing the mesh to make it suitable for parameterizations and improve the parametrization results, for example, by introducing seams. Mesh parts are created which are to be homeomorphic to a disc and the parametrization distortion is reduced [34] and [9]. For a thorough review of the field the reader is referred to a recent survey by Floater and Hormann [13].

Mean-shift has already been used for clustering [2], for segmentation of images [5], video [7,40] and 3D volumetric meshes [32], for motion tracking [6,4] and recognition [41]. It is a powerful tool when used on images, video and volumes and this work extends its applicability to boundary meshes which are so often used in visualization and graphics.

3 Feature Space

The mean-shift operates in a combined feature space that consist of a combination of the domain geometric dimensions $x_1, x_2, \ldots, x_d$ and of various attributes (e.g. functions) defined over the domain $f_1, f_2, \ldots, f_m$. The functions can be scalar or vector functions, they can be predefined such as temperature, density, pressure, and wind-direction, or they can be extracted from the mesh such as gradient, curvature, normals, or centricity (see Figures 2). The choice of using a specific function depends on the application in mind. Any point P in the high dimensional feature-space is defined by combining the geometric coordinates $p = (x_1, x_2, \ldots, x_d)$ of the point on the mesh domain and the values of the functions defined at p: $P = (x_1, x_2, \ldots, x_d, f_1(p), f_2(p), \ldots, f_m(p))$

Working in this high dimensional space consisting of a mix of coordinates - geometric and functional, is not intuitive. The nature, range and scale of the various functional and geometric subspaces can vary considerably, and must be normalized before combining them to feature space. Choosing correct normalization constants is a challenging issue beyond our scope, and often it is task dependent. Nevertheless, the main objective of mean-shift is to find structure in this high dimensional space by searching for dense regions. In practice we normalized the values in each function subspace using its standard deviation. The distance measure $D_\Phi(P,Q)$ between two points P and Q in the high dimensional feature-space Φ is either Euclidean or defined as the sum of Euclidean distances in each subspaces (geometric and attribute).

The functional attributes can be defined either on mesh vertices or on the mesh faces (triangles). When attributes are defined per triangle creating a piecewise constant function over the domain, the result is a piecewise constant feature-space. When attributes are assigned to the vertices and interpolated inside each triangle, the result is a piecewise linear function over the mesh domain, and similar blending is used in feature-space. The choice of blending also depends on the type of filtering selected. When filtering faces we define a representative point for each face at the center of the face and map it to a feature-space point. This imposes feature boundaries which are defined by the edges of the mesh, and a piecewise constant feature space. Filtering vertices imposes the piecewise linear feature space and the feature boundaries are likely to fall inside triangles.

4 mean-shift

The mean-shift procedure starts from some point in feature-space and iteratively follows an estimation of the density function gradient [14] until convergence. The feature-space itself can be regarded as the empirical probability density function. This means that dense regions in feature-space correspond to local maxima in the probability density function (see [5] for details and for convergence proof). Therefore, the mean-shift procedure finds *modes* (i.e. local maxima) in feature-space by moving towards them incrementally. Applying this procedure to a set of given points will create clusters around the modes.

A single step of the mean-shift procedure is performed by defining a neighborhood Ω around the current point in feature-space. This neighborhood is used for the Parzen window density estimation [27]. The procedure calculates the weighted mean of the points that fall inside this neighborhood, and then moves to the mean by shifting the window to be centered around it. Hence the basic step is composed of calculating the mean of all points in the neighborhood Ω around a given point. More specifically, if P is a feature-space point, then the mean $M_h(P)$ of P is defined using a radially symmetric kernel K with radius h in feature-space with a monotonically decreasing profile $g(x)$. For instance,

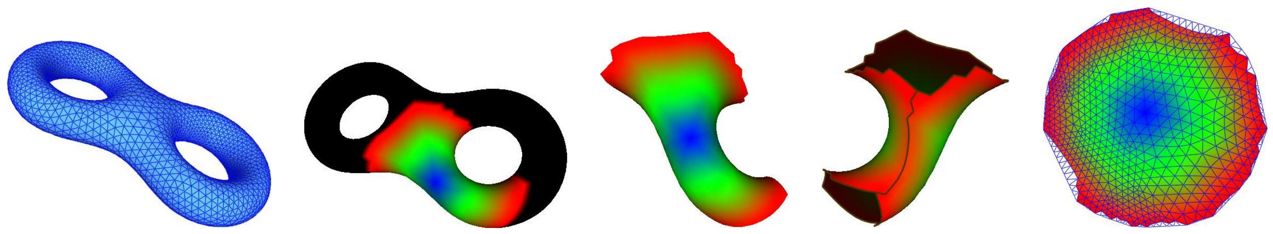

Fig. 3 An overview of the steps of creating local geodesic parametrization. The colors represent the true geodesic distances from the center: from blue (center) to red (periphery). For geodesic mean-shift, instead of adding filler triangles (last image) we pull the vertices back to the convex hull.

we have used normal kernel defined by the common profile $g(x) = e^{-x^2/2}$. Other kernels such as the uniform kernel can be used as well.

For a piecewise constant approximation of feature space we assume each constant region S is represented by a feature space point P_S. The *mean* of a point P using profile $g(x)$ in a neighborhood Ω with radius h around P is defined as:

$$M_h(P) = \frac{\sum_{S\subset\Omega} P_S \cdot g\left(\left|\left|\frac{D_\Phi(P,P_S)}{h}\right|\right|^2\right)\cdot \text{area}(S)}{\sum_{S\subset\Omega} g\left(\left|\left|\frac{D_\Phi(P,P_S)}{h}\right|\right|^2\right)\cdot \text{area}(S)} \tag{1}$$

For a piecewise linear approximation of feature space, we assume we can sample the neighborhood Ω uniformly (see next section) and hence just use the weighted average of all points Q around P:

$$M_h(P) = \frac{\sum_{Q\in\Omega} Q \cdot g\left(\left|\left|\frac{D_\Phi(P,Q)}{h}\right|\right|^2\right)}{\sum_{Q\in\Omega} g\left(\left|\left|\frac{D_\Phi(P,Q)}{h}\right|\right|^2\right)} \tag{2}$$

Nevertheless, there still remains the question of sampling the neighborhood Ω correctly on the mesh, and the definition of D_Φ. Simply using 3D coordinates and Euclidean distances of points on the surface may easily result in mean points which are outside the surface. Furthermore, on surface meshes we need to measure distances correctly on the mesh, by defining the geodesic geometry sub-space inside feature space. Hence, we must constrain the shifting of a point to its mean to remain on the mesh surface (e.g. its domain).

5 Geodesic parametrization

Surface parametrization is a mapping from a 2D domain onto the surface. By definition a parametrization is only possible for an open surface homeomorphic to a disk. Otherwise, the surface must first be cut along at least one edge before it can be mapped onto a planar region. Moreover, with the exception of developable surfaces, general open manifolds cannot be parameterized without distortion. Many parametrization methods assume that the cuts are given and focus on the optimization of the parametrization to minimize some geometric distortion metrics. However, introducing cuts (seams), possibly beyond those necessary to make the surface a topological disk, can reduce this distortion further [35].

For our geodesic mean-shift operator, the elementary operation in equations 1 and 2 is the geodesic distance calculation between points in the neighborhood and the center point. Hence, local parametrization will serve only the central vertex around which the parametrization is built, and the distortion measure must be biased towards this point. Most previous work concentrate on minimizing global distortion of surface attributes such as angles, lengths, areas or even geodesic distances [44].

We define for each vertex its own local *geodesic parametrization* [33] to minimize the distortion of the geodesic distances to all the points in its neighborhood. The parametrization changes while moving from one point to another, creating a moving local geodesic parametrization of the whole surface. Figure 3 gives an overview of our method for a single vertex. The first step in the algorithm is to define the geodesic neighborhood of the vertex. This is done using a front marching algorithm similar to the one used in [21]. Note that if the front meets itself, it merges to produce a new unified front. This means that later there is a need to cut seams in the geodesic neighborhood patch.

The output of this step is a collection of vertices and edges which comprise a vertex's local geodesic neighborhood within a given radius. Even though the mesh is manifold, the computed geodesic neighborhood will not, in general, be homeomorphic to a disk. This is the case when the neighborhood radius is large or the mesh is complex (e.g. fingers of a hand). To create a patch which is homeomorphic to a disk, we wish to ensure that it includes only one boundary loop. This boundary loop will become the boundary of the patch local geodesic map, and define the patch shape.

Based on [34] we connect the different boundary loops and cut the patch to include only one boundary loop. We place the vertices of the boundary loop in order on a 2D circle, whose initial radius is the geodesic neighborhood ra-

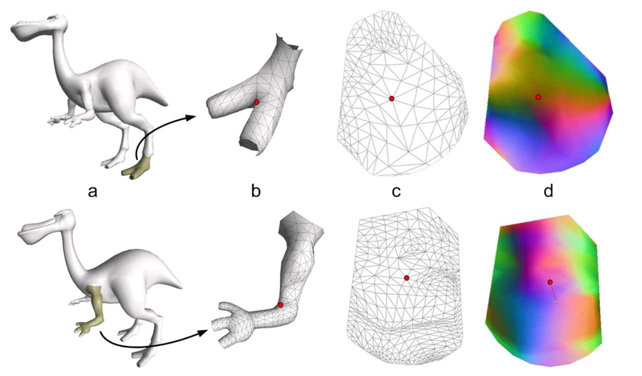

Fig. 4 Two examples of local geodesic parametrization for feature-space neighborhood calculation. The local neighborhood (a-b) of vertices on the mesh is flattened to create Γ_v (c), and the features are mapped to the 2D domain (d). In this example, the normal XYZ values are mapped to RGB.

dius. We move the vertices towards the center to their true geodesic distance from the center. This forms a non-convex shape, which is harder to parameterize. Furthermore, using averaging calculation on non-convex shape may result in a point outside the mesh. Hence, we calculate the convex hull of the 2D shape and pull the vertices back out to lay on the convex hull. Although this may introduce some distortion of the geodesic distances it is usually very small: the original shape is almost convex as the distance to all boundary vertices of the geodesic neighborhood is almost similar. Lastly, mean-value coordinates are used to calculate the parametrization inside the patch [12].

The end result of this process is a 2D patch $\Gamma_v \subset \mathbb{R}^2$ for each vertex v, representing a parametrization of the geodesic neighborhood of v. The connectivity of Γ_v is the same as that of the geodesic neighborhood of v on the mesh. Each vertex and triangle in Γ_v has a mapping onto the mesh creating a mapping from each point $p \in \Gamma_v$ onto the sub-mesh in the geodesic neighborhood of v. This enables a mapping of the features from the mesh to the 2D domain (Figure 4). Due to some distortions, we still store for each 2D triangle in Γ_v the ratio between its area and the original 3D mesh triangle area. This ratio is used as a correction factor in equations 1, and in equation 2 the sample values during averaging are multiplied by this ratio to better represent the weight of the true geodesic neighborhood area.

Performing the geodesic parametrization process on all vertices of a 2-manifold mesh, creates a collection of local geodesic maps. Even when a selected number of vertices are used, the amount of overlap in these maps can be quite large. Hence, the memory footprint of naïve storage is quite large. There is a need to balance the computation speed with storage size and memory efficiency. Our scheme is efficient in terms of storage, but depends on longer computation to reconstruct each patch whenever it is needed. For each patch we store only:

1. The indexes of the vertices in the original mesh which appear in the patch.
2. The set of half edges which represent the seam tree in the patch.
3. The distance and angle from the center vertex to all vertices on the boundary of the patch, taken from the front marching algorithm.

This storage method saves the space of both the connectivity and the geometry information of the patch. During reconstruction we must create a mesh object for the patch, use the connectivity information from the original mesh, cut the patch along the seam tree, and then parameterize the actual patch anew. This does not take more than a few seconds per patch. Nevertheless, while performing mean-shift we use a cache of fully reconstructed patches to amortized the reconstruction costs.

6 Geodesic mean-shift

The principal behind the geodesic mean-shift algorithm is to imagine the surface domain from the perspective of a point laying on the surface. This means that:

1. Distances between points are measured on the surface as geodesic distances, and,
2. Both the neighborhoods and the movements of the mean-shift are constrained to the domain of the surface.

Both these are achieved using the local 2D geodesic parameterization Γ_v around each vertex v (Section 5). Γ_v serves as the geodesic neighborhood which contains the spatial feature space neighborhood Ω around v for the mean calculation. The 2D distances of each point $p \in \Gamma_v$ to the center point v well approximate the corresponding 3D geodesic distances. Furthermore, each point $p \in \Gamma_v$ can be mapped back to the 3D surface using the local parametrization. In effect, by calculating the mean-shift averaging on Γ_v, we constrain the mean to remain on the surface.

The definition of the density function for images uses the Parzen window estimation technique over the feature space as an empirical discrete sampling space. Images can be seen as a uniform sampling of this space. To get uniform geodesic sampling on meshes one should maintain uniform triangulation over all the mesh surface. This can sometimes be achieved using various subdivision and re-meshing techniques. Nevertheless, these techniques can be both difficult and time-consuming, and they alter the mesh connectivity structure which is undesirable when attributes are defined on the mesh. Furthermore, even when the 3D mesh is triangulated uniformly, there is no guarantee that after flattening of the local neighborhoods the uniformity characteristic will be preserved in Γ_v.

Instead, to gain uniform sampling of the local geodesic neighborhoods of points on the mesh we utilize a sampling technique taking advantage of the graphics hardware. Given a point on the mesh, we use the parametrization Γ_v around the vertex v closest to the point. We then rasterize a clipped area inside Γ_v mesh with the radius of the current Parzen window geometric range. To get a sampling of the attribute

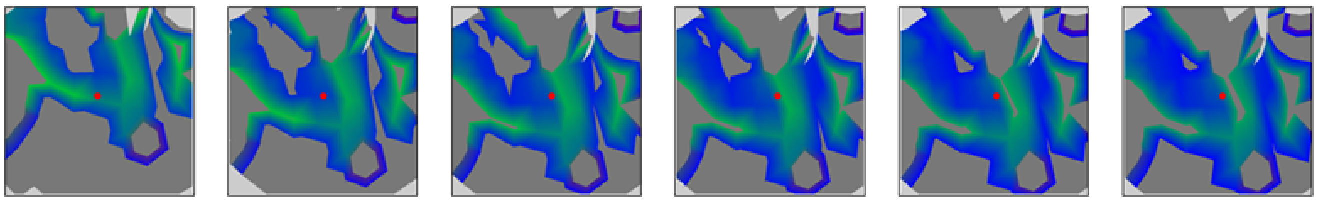

Fig. 5 An example of the mean-shift process using 2D flattening and rasterization. From left to right, each image represents one step of the mean-shift on the surface of the Dino-pet. In each step, the 2D flattened neighborhood around the current position on the mesh (red dot) is rasterized using the feature as one of the color channels. The averaging filters out the attribute values which are out of the range of the point (dark gray regions). Note in this example that the initial steps move more in the spatial sub-space, while the final steps move more in the attribute sub-space (hence changing the filtered attribute areas).

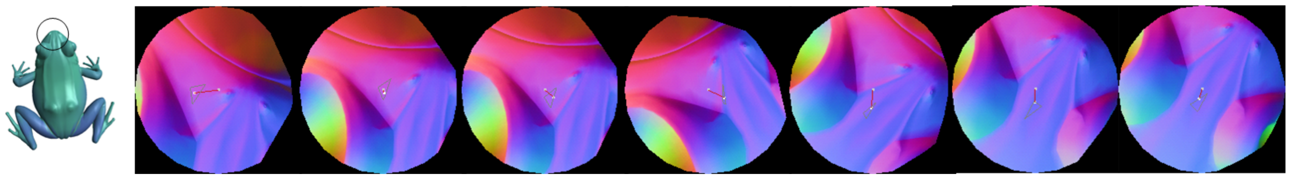

Fig. 6 An example of a sequence until convergence of the mean-shift of one vertex on the frog head with normal feature attribute.

sub-spaces we assign each vertex in the 2D mesh its attribute values in one of the three color channels. Three 8-bit channels can be used simultaneously in OpenGL and up to 16 floating point in some extensions such as CG. The rasterization process creates a bitmap which is a uniform sampling of a 2D window around the vertex. The color channel of each pixel in this bitmap holds the interpolated values of the triangle vertices. This way we create a piecewise linear mapping of the feature space function around the feature-space point. The image is then used for filtering out pixels which are out of the attribute window range and for calculating the new mean (Figure 5 and 6).

7 Examples

To validate the geodesic mean-shift procedure we start with a simple example shown in Figure 7. We begin with a 2D image created out of random sampling of 3 Gaussian distributions. Each gaussian is given its own unique value. The value for each point on the image is chosen as one of the three possible values with probability proportional to the three gaussian values at that point (with a bias towards the larger value). Next, we create a mesh cylinder out of the image and give the vertices the values of the original pixels. Using mean-shift on the 2D image creates three clusters according to the three values, and maps points which are outside the main region to the correct modes. We now employ our geodesic mean-shift procedure to the cylinder and warp the results back to 2D for comparison. As can be seen the same modes were found and the same features structures preserved.

We have tested several surface meshes of various genus, using several possible attributes (Some examples are shown in Figure 8 to 11). The meshes in the examples have up to 20,000 vertices. Local parametrization and flattening created

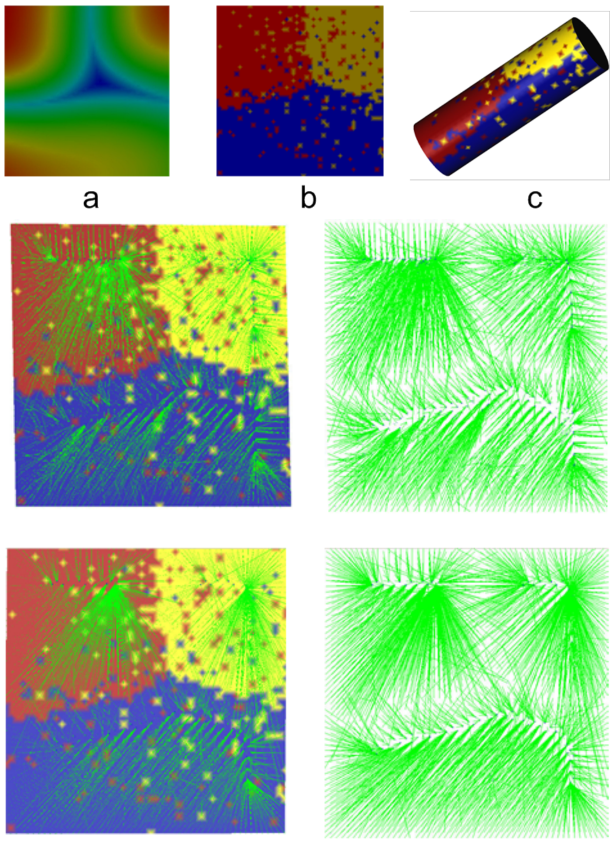

Fig. 7 Validating the Geodesic mean-shift. A Gaussian mixture of three Gaussians (a) is used to sample a random 2D image with three characteristic values (b). The image is then used to create a similar 3D mesh on a cylinder (c). The results of original 2D mean-shift procedure is shown in the middle, and the results of geodesic mean-shift on the cylinder (mapped back to 2D) at the bottom. Each green line indicates the mapping of a point to its mode. As can be seen the results are almost indistinguishable.

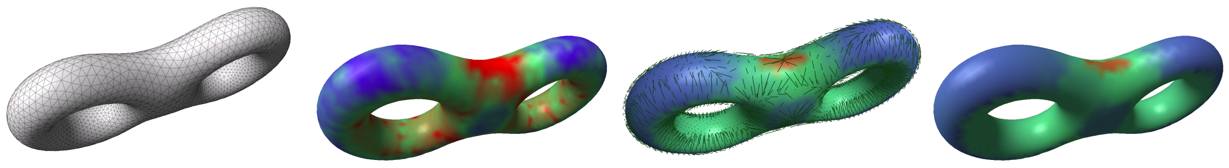

Fig. 8 Mean-shift on a Torus with Gaussian curvature as a one dimensional feature. The first image is the mesh and the second is a color mapping of the Gaussian curvature. In the third image, the lines link each vertex to its mode vertex after mean-shift. The fourth image shows the filtered mesh where the value of each vertex is replaced by the value of its mode.

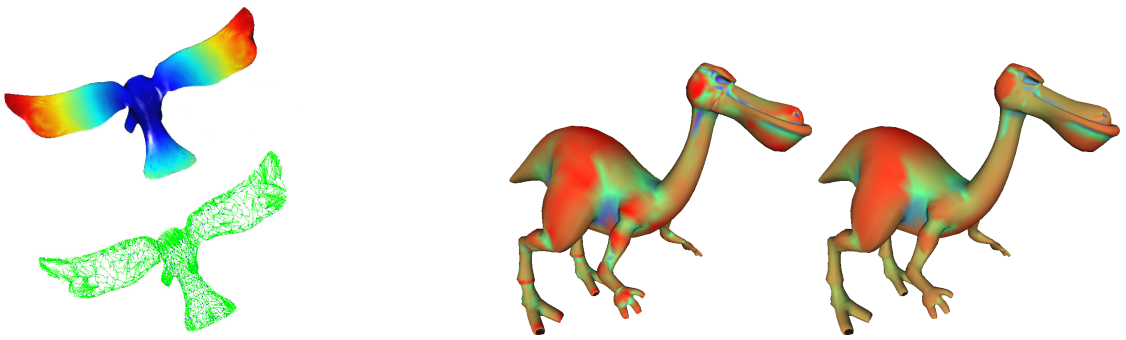

Fig. 9 Left: mean-shift on a bird mesh with centricity field as a one dimensional feature. This feature is gradual over the mesh (top), hence the modes of the vertices are very local (bottom). Right: the dino-pet mesh with a scalar field value defined as $\frac{2}{\pi}\arctan\frac{\kappa_1+\kappa_2}{\kappa_1-\kappa_2}$, Where κ_1 and κ_2 are the principle curvatures and $\kappa_1 \geq \kappa_2$ (see [22]). This type of field enhances the distinction between concave and convex parts of the mesh. The left image (before mean-shift) is more noisy than the right (after mean-shift).

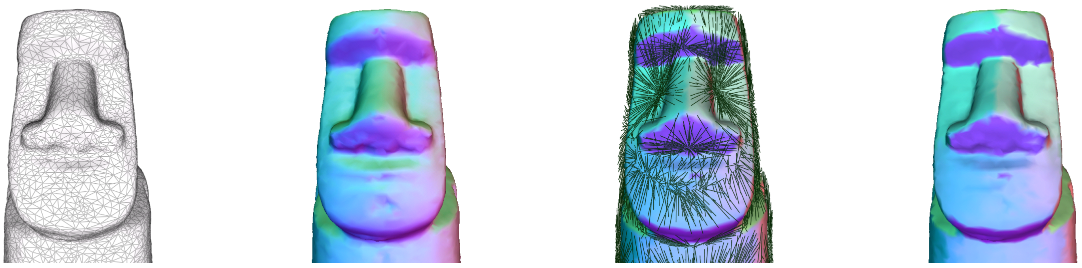

Fig. 10 Mean-shift on a Maoi stone statue with normals XYZ as three dimensional feature. Note the difference in the distribution of normals in the second and last figure. The mean-shift clusters areas of similar normals and enhances the feature lines between them.

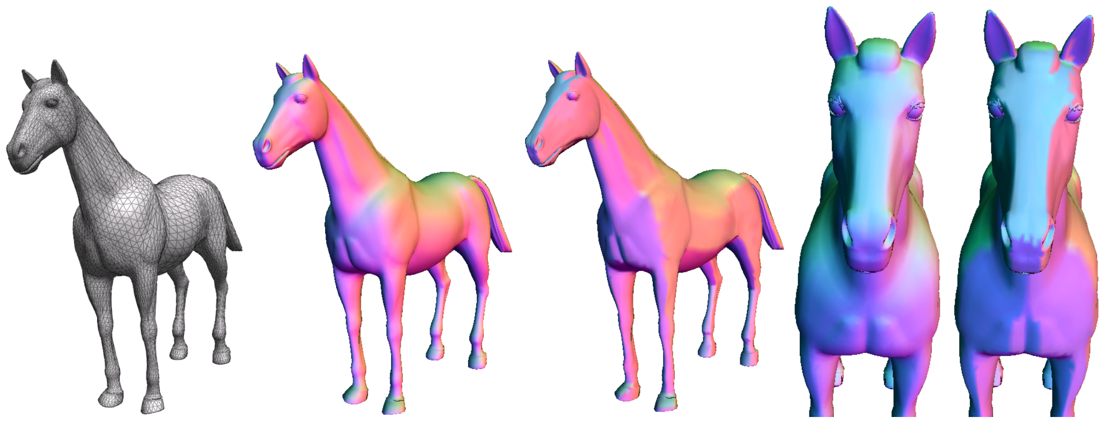

Fig. 11 Mean-shift on a horse mesh with normals: the normals directions become more apparent and uniform and the feature lines are enhanced.

a local 2D map for each vertex in a pre-processing stage taking up to 20 minutes per mesh, using a 2GHZ Pentium-4 machine. For the mean-shift itself we used a kernel radius of around 10% of the bounding box diagonal for each mesh, and 10% of the feature range. The average convergence rate was similar to all previous reports of the mean-shift procedure, and around 7 iterations per vertex. The whole process took between 10 to 60 minutes per mesh in the examples shown.

The choice of feature for mesh analysis depends on the application in mind. We have tested both scalar and vector features and any combination of the features can also be easily applied using our method. One dimensional scalar features defined on the mesh include curvature (Figure 8), centricity, which is the average of the geodesic distance from a vertex to all other vertices (Figure 9, left), and a value disntinguishing concavity and convexity (Figure 9, right). The normal direction (Figures 6, 10, and 11) is a three dimensional (vector) feature value. For scalar fields we use a hue color mapping of the scalar values on the mesh, while for the normals we used a mapping from XYZ to RGB. The mapping of each vertex to its mode is sometimes shown with lines.

These results show that the geodesic mean-shift algorithm on boundary meshes works in a similar manner as mean-shift on rectangle images or on volumes. The procedure converges in a similar convergence rate, while still constrained to remain on the mesh surface. Points are mapped from areas of high feature-gradient energy to areas of low feature-gradient energy. The local modes of the joint density function estimate are found on the mesh surface, and the modes are characteristic of the joint geometric and attribute feature-space.

8 Conclusion and Future Work

This paper defines a generalization of the mean-shift procedure to surface meshes. This generalization is achieved using moving geodesic parametrization and flattening and the definition of continuous linear feature space. The procedure can be used to cluster vertices or faces on boundary meshes and can be utilized for feature extraction or segmentation.

As seen in the given examples, different results can be attained using different features over the mesh. This is sometimes referred to as the feature-selection problem. Often the choice of features is application dependent and it still remains one of the most challenging issues in any analysis, clustering or segmentation algorithm.

In the future, we plan to leverage geodesic mean-shift to achieve more accurate results for segmentation. For instance by combining a clustering algorithm on the filtering results of the geodesic mean-shift. We are also working on improving the efficiency of the process to work on larger and more complex meshes by employing multi-resolution and hierarchical techniques.

References

1. Arabie, R., Hubert, L., DeSoete, G. (eds.): Clustering and Classification. World Scientific Publishers, River Edge, NJ (1996)
2. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Mach. Intell. **17**(8), 790–799 (1995)
3. Cohen-Steiner, D., Alliez, P., Desbrun, M.: Variational shape approximation. ACM Trans. Graph. **23**(3), 905–914 (2004)
4. Collins, R.: Mean-shift blob tracking through scale space. In: Computer Vision and Pattern Recognition (CVPR'03). IEEE (2003)
5. Comaniciu, D., Meer, P.: Mean shift: A robust approach towards feature space analysis. IEEE Trans. Pattern Analysis and Machine Intelligence **24**, 603–619 (2002)
6. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Computer Vision and Pattern Recognition (CVPR'00). IEEE (2000)
7. DeMenthon, D.: Spatio-temporal segmentation of video by hierarchical mean shift analysis. In: Statistical Methods in Video Processing Workshop, SMVP 2002. Copenhagen, Denmark (2002)
8. DeRose, T., Kass, M., Truong, T.: Subdivision surfaces in character animation. In: ACM Computer Graphics, Proc. SIGGRAPH 1998, pp. 85–94 (1998)
9. Erickson, J., Har-Peled, S.: Optimally cutting a surface into a disk. In: Proceedings of the 18th Annual ACM Symposium on Computational Geometry, pp. 244–253 (2002)
10. Faugeras, O.D., Hebert, M.: The representation, recognition, and positioning of 3-d shapes from range data. In: T. Kanade (ed.) Three-Dimensional Machine Vision, pp. 301–353. Kluwer Academic Publishers (1987)
11. Floater, M.: Parametrization and smooth approximation of surface triangulations. Computer Aided Geometric Design **14**, 231–250 (1995)
12. Floater, M.: Mean value coordinates. Computer Aided Geometric Design **20**, 19–27 (2003)
13. Floater, M.S., Hormann, K.: Surface parameterization: a tutorial and survey. In: N. Dodgson, M.S. Floater, M. Sabin (eds.) Advances on Multiresolution in Geometric Modelling. Springer-Verlag, Heidelberg (2005)
14. Fukunaga, K., Hostetler, L.D.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transaction on Information Theory **IT-21**, 32–40 (1975)
15. Garland, M., Willmott, A., Heckbert, P.: Hierarchical face clustering on polygonal surfaces. In: Proc. ACM Symposium on Interactive 3D Graphics (2001)
16. Gordon, A.D.: Hierarchical classification. In: R. Arabie, L. Hubert, G. DeSoete (eds.) Clustering and Classification, pp. 65–105. World Scientific Publishers, River Edge, NJ (1996)
17. Guralnik, V., Karypis, G.: A scalable algorithm for clustering protein sequences. In: Workshop on Data Mining in Bioinformatics (2001), pp. 73–80 (2001)
18. Inoue, K., Takayuki, I., Atsushi, Y., Tomotake, F., Kenji, S.: Face clustering of a large-scale cad model for surface mesh generation. Computer Aided Design **33**(3) (2001). The 8th International Meshing Roundtable Special Issue: Advances in Mesh Generation
19. Kalvin, A.D., Taylor, R.H.: Superfaces: Polygonal mesh simplification with bounded error. IEEE Computer Graphics and Applications **16**(3) (1996)
20. Katz, S., Tal, A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. ACM Transactions on Graphics (Proceedings SIGGRAPH 2003) **22**(3), 954–961 (2003)
21. Kimmel, R., Sethian, J.: Computing geodesic paths on manifolds. pp. 8431–8435 (1998)
22. Koenderink, J., van Doorn, A.: Surface shape and curvature scales. Image and vision computing **10**, 557–565 (1992)
23. Kraevoy, V., Sheffer, A.: Cross-parameterization and compatible remeshing of 3D models. ACM Trans. Graph. **23**(3), 861–869 (2004)
24. Levy, B., Mallet, J.L.: Non-distorted texture mapping for sheared triangulated meshes. In: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pp. 343–352 (1998)

25. Levy, B., Petitjean, S., Ray, N., Maillot, J.: Least squares conformal maps for automatic texture atlas generation. In: ACM Computer Graphics, Proc. SIGGRAPH 2002, pp. 362–371 (2002)
26. Mangan, A.P., Whitaker, R.T.: Surface segmentation using morphological watersheds. In: Proc. IEEE Visualization 1998 Late Breaking Hot Topics (1998)
27. Parzen, E.: On estimation of a probability density function and mode. Annals of Mathematical Statistics **33**, 1065–1076 (1962)
28. Pulla, S., Razdan, A., Farin, G.: Improved curvature estimation for watershed segmentation of 3-dimensional meshes (2001). Manuscript
29. Roberts, S.J.: Parametric and non-parametric unsupervised cluster analysis. Pattern Recognistion **30**, 327–345 (1997)
30. Sander, P., Wood, Z., Gortler, S., Snyder, J., Hoppe, H.: Multi-chart geometry images. In: Proc. Eurographics Symposium on Geometry Processing 2003, pp. 146–155 (2003)
31. Schreiner, J., Asirvatham, A., Praun, E., Hoppe, H.: Inter-surface mapping. ACM Trans. Graph. **23**(3), 870–877 (2004)
32. Shamir, A.: Feature-space analysis of unstructured meshes. In: Proceedings IEEE Visualization 2003, pp. 185–192. Seattle, Washington (2003)
33. Shapira, L., Shamir, A.: Local geodesic parametrization: An ant's perspective (2005). Submitted for publication
34. Sheffer, A.: Spanning tree seams for reducing parameterization distortion of triangulated surfaces. In: Proceedings of the International Conference on Shape Modeling and Applications 2002 (SMI'02), pp. 61–66 (2002)
35. Sheffer, A., Hart, J.: Seamster: Inconspicuous low-distortion texture seam layout. In: Proc. IEEE Visualization 2002, pp. 291–298 (2002)
36. Sheffer, A., de Sturler, E.: Surface parameterization for meshing by triangulation flattening. In: Proceedings of the 9th International Meshing Roundtable, pp. 161–172 (2000)
37. Shlafman, S., Tal, A., Katz, S.: Metamorphosis of polyhedral surfaces using decomposition. Computer Graphics forum **21**(3) (2002). Proceedings Eurographics 2002
38. Sorkine, O., Cohen-Or, D., Goldenthal, R., Lischinski, D.: Bounded-distortion piecewise mesh parameterization. In: Proc. IEEE Visualization 2002 (2002)
39. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proc. of the sixth international Conference of Computer Vision, pp. 839–846 (1998)
40. Wang, J., Xu, Y., Shum, H.Y., Cohen, M.F.: Video tooning. ACM Trans. Graph. **23**(3), 574–583 (2004)
41. Yu, B.: Recognition of freehand sketches using mean shift. In: Proceedings of the 8th international conference on Intelligent user interfaces, pp. 204–210. ACM Press (2003)
42. Zhuang, X., Huang, Y., Palaniappan, K., Zhao, Y.: Gaussian mixture density modeling: Decomposition and applications. IEEE Transactions on Image Processing **5**(9), 1293–1302 (1996)
43. Zigelman, G., Kimmel, R., Kiryati, N.: Texture mapping using surface flattening via multi-dimensional scaling. IEEE Transactions on Visualization and Computer Graphics **8**(2), 198–207 (2002)
44. Zigelman, G., Kimmel, R., Kiryati, N.: Texture mapping using surface flattening via multidimensional scaling. IEEE Transactions on Visualization and Computer Graphics **8**(2), 198–207 (2002)