Scale Space Based Feature Point Detection on Surfaces

Markus Schlattmann Patrick Degener Reinhard Klein

Universität Bonn
Institut für Informatik II - Computergraphik
D-53117 Bonn, Germany
{markus, degener, rk}@cs.uni-bonn.de

ABSTRACT

The detection of stable feature points is an important preprocessing step for many applications in computer graphics. Especially, registration and matching often require feature points and depend heavily on their quality. In the 2D image case, scale space based feature detection is well established and shows unquestionably good results. We introduce a novel scale space generalization to 3D embedded surfaces for extracting surface features. In contrast to a straightforward generalization to 3D images our approach extracts intrinsic features. We argue that such features are superior, in particular in the context of partial matching. Our features are robust to noise and provide a good description of the object's salient regions.

Keywords: Feature Detection, Intrinsic, Surface, Scale Space.

1 INTRODUCTION

The identification of salient geometric features is crucial for many 3D applications in computer graphics. In morphing applications a feasible mapping between two objects is computed, where salient regions should be mapped on corresponding regions, for example eyes on eyes (regarding mappings between animals). Other applications such as feature based registration or matching rely on the computation of suitable features, too. Thereby, two major requirements on the features should be satisfied in order to support practical results. First, the features have to be robust to marginal changes or noise, because otherwise two similar objects could have two very different feature sets resulting in wrong correspondences. Second, the extracted features have to be distinctive, they should correspond to regions that are characteristic for the particular object or its class of objects. If the features describe non-characteristic regions it would often be impossible to distinguish very different objects.

Having robust and distinctive features at hand, a feature driven and therefore plausible matching between similar objects or parts of objects is possible. Unfortunately, the scaling of similar objects is often different. For example matching an adult and a little child based on features with a fixed scale would mostly fail. While this case can be solved by a simple scaling based on the object size, it becomes more complicated if the matching is partial (e.g. parts of one object are missing). In this case, the scale can only be computed from local properties. Therefore we use a scale space based approach, which, related to well known 2D image based methods, extracts feature points with an associated scale parameter. This scale parameter indicates the extent of the inherent local shape, which enables scale invariant matching. Additionally, a partial matching of objects in the same scale can as well be improved by simply rejecting correspondences between features of different extends.

2 RELATED WORK

Several approaches introduced techniques to find features on 3D surfaces, often used in the context of shape matching or shape retrieval applications. The great amount of literature in this area makes it practically impossible to give a full review on these methods. Therefore we focus on previous work most closely related to our method and refer the reader to state of the art reports for broad overviews in related areas as for instance [TV04], [BKS$^+$05] and [IJL$^+$05].

The first two methods we want to mention here follow the idea of subdividing the surface into small regions and then selecting the most distinctive ones as a representative feature set in order to match or retrieve 3D objects. Shilane and Funkhouser [SF07] first sample a 3D surface by a set of random points. For each point, a spherical descriptor is evaluated in four different radii and the descriptor difference of all pairs is computed to produce a ranked list with respect to a set of equally processed and already classified objects. These lists are then analyzed to produce measures of distinctiveness for a specific class of objects and their descriptors. Finally, a small set of most distinctive features is extracted to represent the object.

A partial surface matching method based on local descriptors was introduced in [GCO06]. The surface is divided into small regions, whose local shapes can be well approximated by quadrics. These regions are used as descriptors and the most salient ones are chosen for the partial matching process. Unfortunately, this method seems to be sensible to noise, because of the dependency of the extracted surface regions on local curvature. Moreover, no scale parameter is extracted, so partial matching of different scaled objects is not possible.

Other approaches aim at extracting the topological structure of an object in order to perform matching or retrieval. In this context Tam and Lau [TL07] introduced a novel method for the retrieval of deformable 3D models. They extract topological points and rings, which are identified by solving a flow and transportation (EMD) problem, which is based on the construction of reeb graphs. While this method shows great results for retrieving articulated shapes as a whole, it is unclear how to generalize it to the partial matching context.

Several so called multi-scale methods extract features of different sizes to gain more geometric information. For example in Clarenz et al. [CRT04] feature points and lines are extracted by performing a local momentum analysis of the surface. To detect features of different scales they adopt variable neighborhood sizes resulting in increased robustness to noise. Unfortunately, it is not suitable to extract the unique scale of a feature, because the neighborhood size parameter is specified manually.

The last category of geometric feature point extraction methods we want to mention here are scale space based approaches. These methods extract salient features with an incorporated scale parameter, which indicates the size of the inherent structure. Li and Guskov [LG05] introduced a novel registration method for point sampled surfaces. They detect feature points on the basis of the scale space theory of Lindeberg [Lin98]. Thereby the surface is smoothed with increasing neighborhood sizes (euclidian balls) using a least squares formulation. This method works well with simple objects, however considering more complex objects this approach will lead to unwanted behavior. This is because euclidian neighborhoods of large sizes are used and therefore often parts of the object are contained, that are far away from the feature in the geodesic sense and should actually not have influence on the feature point. Furthermore the used formulation does not correspond to the scale space theory, so the meaning of the extracted scale parameter is unclear.

In [WNK06] and [NDK05] a partial matching between 3D objects is performed using volumetric scale invariant feature points. To extract these points a 3D scale space of the binary (either inside or outside the object volume) 3D voxel image is built and blob features are detected in the object volume. For each feature a descriptor is computed and a sub part matching is performed. While this approach extracts scale invariant features, these features are not intrinsic and therefore much less distinctive. For example considering a very elongated part of an object (e.g. a finger of a hand), the volumetric blob will only describe the thickness of the tip, which does not change if the elongation has changed. However an appropriate intrinsic feature point with associated scale will describe a combination of the length and the thickness, which delivers a superior description of this object part and its size. Furthermore, at tapered tips this method would miss this feature completely, because no blob would have been found.

3 GENERAL SETUP AND NOTATION

Our objective in this paper is to extract scale invariant feature points on a 3D model. These features are intrinsic, because they depend only on the surface. In the following we assume that the object is represented as a closed two manifold surface. In addition to that, we consider only objects with genus zero. The surface is a triangulated mesh M with $M = \{V, E\}$, where $V = \{v_i | v_i \in \mathbb{R}^3, i = 1, ..., |V|\}$ is a set of vertices and $E = \{e_{ij}\}$ the set of edges which connect the vertices. A face is given, if a cycle of three edges e_{ij},e_{jk} and e_{ki} exists. For each vertex v_i, a normal n_i can be computed.

4 BLOB FEATURES IN 2D

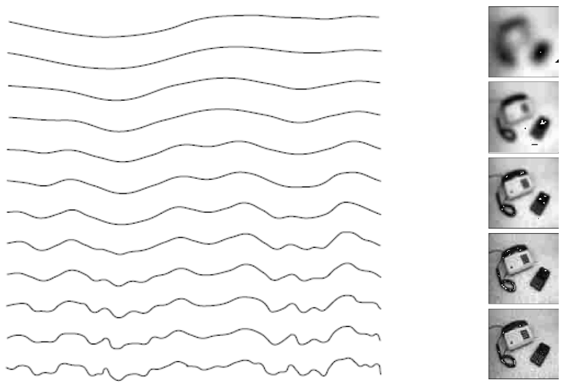
The detection of feature points is well established in 2D image applications. Many feature based matching methods, as surveyed in [ZF03], have shown great practical utility. Especially scale space based techniques [MTS$^+$05] are known for their performance and robustness and therefore often used in practice. A scale space or representation over scales is computed by successive smoothing an input signal to a space consisting of the smoothed signals. In this space, the scale parameter determines the magnitude of the smoothing of the input signal. Figure 1 shows two input signals (bottom), that are iteratively smoothed to obtain a scale space. A scale space of a function $f : \mathbb{R}^D \to \mathbb{R}$ is defined as follows: If a continuous signal f is given, then a scale space $L : \mathbb{R}^D \times \mathbb{R}_+ \to \mathbb{R}$ of f is defined as the solution of the heat diffusion equation

$$\partial_t L = \frac{1}{2}\nabla^2 L = \frac{1}{2}\sum_{i=1}^{D} \partial_{x_i x_i} L, \qquad (1)$$

with $L(\cdot, 0) = f(\cdot)$. This scale space can be computed by convolution of $f(\cdot)$ with a Gaussian kernel g:

$$L(\cdot, t) = g(\cdot, t) \otimes f(\cdot), \qquad (2)$$

with $g : \mathbb{R}^D \times \mathbb{R}_+ \backslash \{0\} \to \mathbb{R}$. Note that the Gaussian kernel is the unique kernel to solve the diffusion equation, what was shown in [Koe84, JWBD86].



(a) (b)

Figure 1: An input signal (bottom) is iteratively smoothed to obtain a scale space. a) One dimensional. b) Two dimensional with marked extrema. [Wit83]

To detect *scale invariant* blob features, Lindeberg [Lin98] used a scale-normalized Laplacian of Gaussian (LoG) function $t\nabla^2 L$ to detect features in the scale space. Scale invariance means, that if an image is scaled with a certain factor, then its features corresponding to features of the non-scaled image will be detected in scales, which are multiplied with the same factor. In Figure 2 two exemplary scale invariant feature points are shown with their signatures, detected in the scale-normalized LoG.

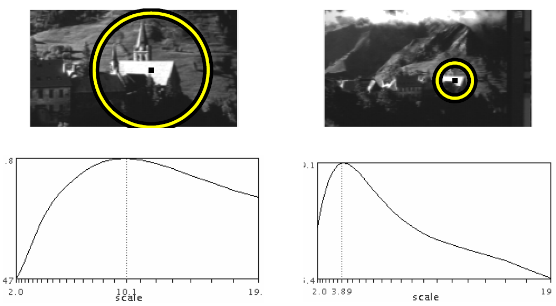


Figure 2: Top row shows two images taken with different zoom. Bottom row shows the responses of the Laplacian over scales. The ratio of scales corresponds to the scale factor (2.5) between the two images. The radius of displayed regions in the top row is equal to 3 times the selected scales. [MTS+05]

For the case of 2D images Lowe [Low04] introduced a so called difference of Gaussian representation (DoG) of f, defined as follows:

$$\begin{aligned} DoG(x,t) &= (g(x,kt) - g(x,t)) \otimes f(x) \\ &= L(x,kt) - L(x,t). \end{aligned} \tag{3}$$

The initial image is incrementally convolved with Gaussians to produce images separated by a constant factor k in the scale space. Adjacent images are subtracted to produce the so called difference of Gaussian images. For the discrete case, beginning with a constant σ_0 (e.g. $\sigma_0 = 1$), the σ_i are obtained as follows:

$$\sigma_i = k^i \sigma_0, \tag{4}$$

where $t = \sigma^2$. This results in an exponential time step. To be able to find all extrema, the factor k should be small enough. Lowe [Low04] used values from the interval $(1;\sqrt{2}]$. Depending on the magnitude of σ_0, more or less initial scales of L are excluded for building the *DoG*. Lowe [Low04] used this representation to approximate the scale normalized Laplacian of Gaussian.

A feature point is extracted, if a pixel in a level of the DoG has an extremal value with respect to its spatial neighbors in the same scale as well as to its and their temporary neighbors in adjacent scales. The information about the scale a feature was detected in is a great advantage, because the scale indicates the size of the structure the feature point describes. In addition to that, the feature points of two images of different resolutions can be compared in an appropriate manner, because of the scale invariance property.

Following this idea, we generalized the scale space and the feature extraction from the 2D image case to the case of triangulated two manifold surfaces in 3D. We use a diffusion flow to derive the sequence of smoothed surfaces and use the vertex movements as a measure similar to the DoG-values in order to extract feature points as well as their scale.

5 GENERALIZATION TO SURFACES

To simulate the diffusion equation (see Equation 1), we use a surface diffusion flow to iteratively smooth the model and to obtain a set of smoothed surfaces that constitute our scale space.

In this section we first describe the mean curvature flow and some of its properties. Furthermore, we give the discretisation used in our implementation and finally, the definition of our feature points is introduced.

5.1 Building the Scale Space

In the image case usually a Gaussian kernel is used to generate the representation over scales. That is possible because it exists a global parameterization invariant over all scales. However, in the case of two manifold surfaces such a parameterization is generally not defined. But nevertheless, a local parameterization for each vertex in each scale is calculable. Therefore an iterative flow is utilizable to simulate a similar diffusion process.

Averaged Mean Curvature Flow The ordinary mean curvature flow is defined as follows:

$$\frac{\partial v_i}{\partial t} = -H_i n_i, \tag{5}$$

where H_i is the mean curvature at vertex v_i. $\frac{\partial v_i}{\partial t}$ is the position increment vector of vertex v_i so the new position results in $\widetilde{v_i} = v_i + \frac{\partial v_i}{\partial t}$. That means, a vertex v_i is moved in direction of its normal n_i with the magnitude of the mean curvature $H = \frac{1}{2}(\kappa_{min} + \kappa_{max})$, where κ_{min} and κ_{max} denote the principal curvatures. A vertex on a convex region will move inwards, whereas a vertex on a concave region will show an outward movement. At a saddle point, the minimal curvature is negative, while the maximal curvature is positive, so the direction of the movement depends on their magnitudes.

The mean curvature flow is known to shrink volume. Thus, a closed surface with genus zero will evolve into an infinitesimally small sphere (see Figure 3).

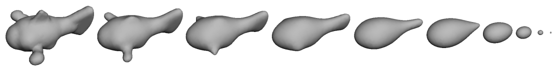


Figure 3: Ordinary mean curvature flow evolves objects to an infinitesimal small sphere.

Therefore we use a modification of the ordinary mean curvature flow: the averaged mean curvature flow, which is defined as follows:

$$\frac{\partial v_i}{\partial t} = -(H_i - \sum_{v_j \in M} \frac{H_j}{|V|})n_i. \qquad (6)$$

The result is a volume preserving flow as shown in Figure 4. Whereas the averaged mean curvature flow is more stable than the ordinary one, it still suffers from one deficiency. If an object has a long thin limb, the flow will trench it after a few steps as shown in Figure 5. However, with a little variation in the thickness, it is possible, that the object is not fragmented. This results in big variations of the feature detection, so that the computed features for such objects are not robust. For this reason, it is only useful for restricted types of objects. Therefore, in our work, we use and compare only objects, that do not cause fragmentations. Note that such a fragmentation can be detected in the smoothing process by checking if local mesh triangles are degenerated to line segments or points.

Figure 4: Averaged mean curvature flow evolves objects to a sphere with the same volume.

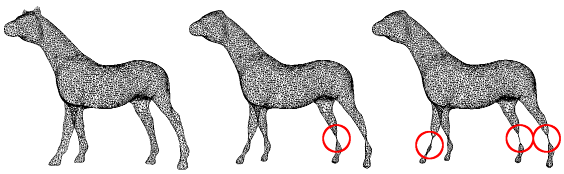


Figure 5: Mean curvature flow trench thin limbs after a few steps.

Another approach to derive and smooth a surface from polygonal data to multiple scales is done in [SOS04]. By using a constrained moving least-squares formulation a surface can be generated, which approximates the input, whereas features with a specified size are smoothed away. Unfortunately, if the surface nearly touch itself, it will accrete at this point, so that marginal differences of the surface could result in a highly different behavior of this smoothing process. For this reason, this formulation of a smoothing of a surface is not usable to replace the mean curvature flow in order to solve the problem of fragmentation.

Discretisation In the following the implementation details for the iterative computation of the flow are provided. The principal curvatures are computed by first locally approximating the surface with a quadratic function and then computing the eigenvalues of its hessian, which correspond to the principal curvatures. The sampling of the local neighborhood is obtained via the Dijkstra-Algorithm, it consists of the n nearest vertices v_{i_k} of vertex v_i.

To fit a quadratic function in the collected points, first the sampled points v_{i_k} have to be transformed onto the tangent plane of v_i. For that purpose two arbitrary orthonormal vectors o_1 and o_2, lying in the plane with normal n_i, are computed. Then the sample points are transformed to points q_k as follows:

$$q_k = ((v_{i_k} - v_i) * o_1, (v_{i_k} - v_i) * o_2). \qquad (7)$$

To get the coefficients $c_l \in \mathbb{R}$, the basis $\{B_l(\xi_1, \xi_2)\}_{l=1}^5 = \{\xi_1, \xi_2, \frac{1}{2}\xi_1^2, \xi_1\xi_2, \frac{1}{2}\xi_2^2\}$ of the quadratic functions (without constant coefficient) is used to set up the following system of equations:

$$\sum_{l=1}^{5} c_l B_l(q_k) = (v_{i_k} - v_i)n_i, \qquad k = 1, \ldots, n. \qquad (8)$$

With $A = (B_l(q_k))_{k=1,l=1}^{n,5} \in \mathbb{R}^{n\times 5}$ and $C = (A^T A)^{-1} A^T \in \mathbb{R}^{5\times n}$ is its pseudo inverse matrix, it can be written as

$$[c_1, \ldots, c_5]^T = C[(v_{i_1} - v_i)n_i, \ldots, (v_{i_n} - v_i)n_i]^T. \qquad (9)$$

This way the coefficients of the quadratic function $f(x,y) = c_1 x + c_2 y + c_3 x^2 + c_4 xy + c_5 y^2$ can be calculated and by computing the eigenvalues of the function's hessian matrix we get the principal curvatures. This scheme is based on the quadratic fitting technique from Xu [Xu04].

Remeshing Since geometry changes greatly during smoothing, the mesh has to be adopted, in order to obtain a mesh with neither too large nor too small or narrow triangles. To this end we use flips, collapses and splits. After each smoothing step the following tasks are executed in sequence:

1. Flip all edges e_{ij}, if the resulting edge is shorter than $\|v_i - v_j\|$ and the angle between the normals of the two adjacent facets of e_{ij} is smaller than three degrees. This improves the structure of the mesh without adding or deleting a vertex.

2. Collapse all edges e_{ij}, if their lengths are below one fifth of the average edge length. This avoids too small triangles.

3. Split all edges e_{ij}, if their lengths are above five times of the average edge length or if the roundness of one of the adjacent triangles is above 1.5. The roundness is defined as the ratio between the radius of the circumcircle and the length of the shortest edge of the triangle. This avoids too big or narrow triangles.

The movement of the vertices in one smoothing step is very small, so one iteration after each smoothing is sufficient. Additionally, we assume the initial meshes to have a structure, which does not make such an remeshing operation necessary.

5.2 Scale Space Signatures

To define the scale space signatures, we first need to formally define our scale space L. Because we are using an explicit scheme, the time step between two scales has to be constant and not too large. If the sample rate is higher, the time step in the smoothing process should be smaller, because otherwise oscillations and other singularities would arise. Especially the exponentially enlargement would cause those problems. For this, we first build a discrete scale space as follows:

$$\begin{aligned} L_D(v,j) &= \sum_{i=0}^{j} d_i(v), \quad j \in \mathbb{N}, \qquad (10)\\ d_i(v) &= sign(v,i)\|\frac{\partial v^i}{\partial t}\| \\ sign(v,i) &= \begin{cases} -1 & , \ \text{if } \langle \frac{\partial v^i}{\partial t}, n^i \rangle < 0 \\ 1 & , \ \text{else} \end{cases} \end{aligned}$$

with v^i is the vertex v in scale i ($v^0 = v$) and n^i its normal in this scale. $d_i(v)$ are the signed distances between two scale levels i and $i+1$ of vertex v. To get an approximation to a continuous scale space with scale level σ, we use the discrete values with

$$L(v,\sigma) = L_D(v,\lfloor\sigma\rfloor) + (\sigma - \lfloor\sigma\rfloor)d_{\lceil\sigma\rceil}(v), \quad \sigma \in \mathbb{R}. \qquad (11)$$

Now, we define analogously to the discrete difference of Gaussian representation of Lowe [Low04]:

$$\begin{aligned} D(v,j) &= L(v,\sigma_{j+1}) - L(v,\sigma_j), \quad j \in \mathbb{N}, \qquad (12)\\ \sigma_j &= k^j \sigma_0. \end{aligned}$$

σ_0 depends on the constant smoothing step, that is used to smooth the surfaces. If the resolution is high, the step has to be smaller than for a mesh with a lower resolution. Moreover, in order to subdivide each octave of σ_0 to sixteen steps, we used $k = 2^{\frac{1}{16}}$.

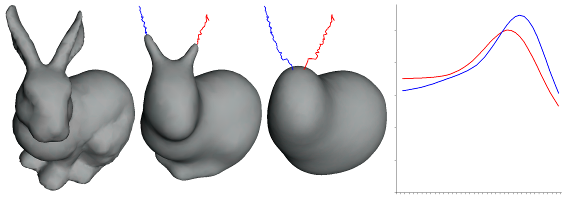


Figure 6: (Left) The trajectories of two vertices on the ears of the bunny. (Right) The scale space signatures (smoothed) of the trajectories.

As a signature S of a vertex v we now use the vector $S = \{D(v,0),...,D(v,m-1)\}$, where m denotes the maximal computed scale. In Figure 6 the trajectories and signatures of two vertices are shown.

5.3 Feature Points

In the application of feature detection we need features which provide a sufficient description of the surface and stays nearly the same, if the object changes marginally.

In our case, we compute feature points as extrema on *extremum paths* as analogously done in [Lin98]. An extremum path r is a sequence of extremal vertices over the scales. That means, the vertices $r(i)$ of the maximum path r have locally maximal signature values in all scales $i = 1,\ldots,l$:

$$D(r(i),i) \geq \max_{v_k \in N^i(r(i))} (D(v_k,i)), \qquad (13)$$

where v_k are the neighbors of $v = r(i)$ in scale i and l is the length of the path. Note that a vertex v has a different position depending on the scale that is considered. Is $d^i_{geo}(v,w)$ the geodesic distance of two vertices in scale i and the signature values of vertices v_j are maximal in respect to their neighbors in this scale, then the following constraints have to be satisfied:

$$\begin{aligned} \forall v_j: \quad d^i_{geo}(r(i-1),r(i)) &\leq d^i_{geo}(v_j,r(i)) \quad \text{and} \\ d^i_{geo}(r(i-1),r(i)) &\leq d^i_{geo}(v_j,r(i-1)), \\ & i = 1,\ldots,l. \qquad (14) \end{aligned}$$

Note that the length l of a path r depends on whether a following maximum exists or not. The computation of the minimum paths is analogously done. An extremum path always begins in the first scale and ends if no following extremum exists.

Now, we detect v as a feature vertex in scale i, if it is included in a maximum/minimum path r with $r(i) = v$ and if the value $D(r(i),i)$ is maximal/minimal with respect to its neighbors $r(i-1)$ and $r(i+1)$.

5.4 Reducing Noise

To reduce noise due to remeshing (because of its local changes in the triangulation), a filtering over the mesh (see Figure 7) on the one hand and a filtering over the signatures of the extremum paths (see Figure 8) on the other hand is done with Gaussian kernels. The standard deviation σ of the first Gaussian kernel (two dimensional) is set in dependency of the average edge length in the mesh. This is a good choice, because normally the higher the resolution (corresponds to the average edge length) of a mesh, the smaller are the structures in the mesh that can be modeled and the more feature points should and can be extracted. In our application we took a width of twice the average edge length. The standard deviation of the second Gaussian kernel (one dimensional), used to smooth the signatures, is set to four.

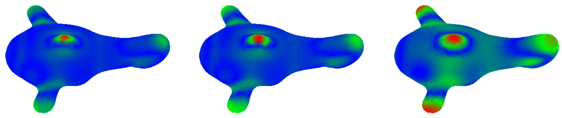


Figure 7: A fish with relatively colorcoded differences. (Left) Unfiltered. (Middle) Filtered with $\sigma = 2$. (Right) Filtered with $\sigma = 4$.

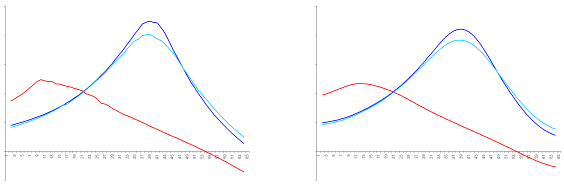


Figure 8: The scale space signatures of three extremum paths of the fish model. (Left) Unfiltered. (Right) Filtered with $\sigma = 8$.

5.5 Eliminating Unstable Features

If a feature point describes a ridge or ravine like structure of the object, often its position is not well determined, because the vertices along this structure have very similar DoG-values. For this reason, Lowe [Low04] introduced the *hessian condition*. This condition rejects such feature points by thresholding the ratio of its eigenvalues. Therefore, the eigenvalues λ_{max} and λ_{min} of the hessian matrix H in respect of the difference of Gaussian values

$$H = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{pmatrix} \tag{15}$$

are computed. Now, if the ratio $\frac{\lambda_{min}}{\lambda_{max}}$ is above 0.5, the point is not taken as a feature. Additionally, features are rejected, if their eigenvalues of H have different signs. Because of this threshold all unstable feature points can be removed. Analogously to the image case, we compute the hessian matrix of a feature point in its scale with an radius proportional to its scale. By this, we get a good indicator for figuring out, whether a feature has an unstable position.

6 RESULTS

In this section, several examples of our feature detection method are presented. For all examples, the same thresholds and widths of the Gaussian kernels to smooth the DoG-values are used.

In all following figures, the feature points detected as a maximum are printed in red, while those detected as a minimum are printed in blue. The signatures of the extremum paths are printed in accordant colors. A feature point is illustrated as a circle with a radius proportional to the scale the feature was detected in. Thereby, the object is shown in the scale of the feature points.

The computation times for the following examples ranged from 30 seconds (for approx. 1400 vertices) to 20 minutes (for approx. 5000 vertices). For meshes larger than 10000 vertices a computation time of more than 2 hours is needed. Therefore, we decided to simplify large meshes in a preprocessing step. To this end a curvature driven simplification is used in order to preserve small features.

6.1 Differently Scanned Objects

To show the robustness by extracting feature points of differently sampled models, the features of two ants with different resolutions are shown in Figure 11. It can be seen, that the same features are extracted, and only the signatures differ marginally.

6.2 Similar Objects

To demonstrate the robustness of our method for pose invariance, we applied our technique on three postures of a hand. The results in Figure 12 show a great attitude in this case.

6.3 Other Examples

The third feature point of the vase in Figure 9 shows, that important features are found, which probably would not be found by other methods.

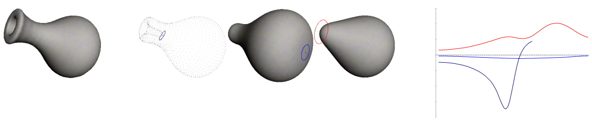


Figure 9: Feature points and signatures of a vase. (Left) Original model (approx. 1500 vertices). (Middle) Smoothed object in scales of the features. (Right) Signatures.

In the feature detection process for the Max Planck head in Figure 13 a lower threshold (0.35) is used for performing the hessian condition, because otherwise the nose and the ears would not have been extracted.

Unfortunately, the problem of choosing the most appropriate threshold arises, so we think, that in a practical application the ratio of the eigenvalues should better be used as a confidence of a feature than for thresholding.

Last but not least we applied our method also on the Stanford Bunny. The results are shown in Figure 10.

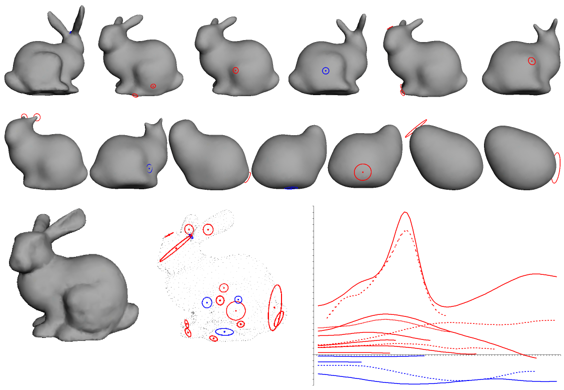


Figure 10: Feature points and signatures of the Stanford Bunny. (Top) Smoothed object in scales of the features. (Bottom) (Left) Original model (approx. 3100 vertices). (Middle) All feature points. (Right) Signatures.

7 CONCLUSIONS AND FUTURE WORK

Robust feature points are needed for many applications, as for instance matching and morphing. Based on approved methods for the image case, we introduced a novel technique for the extraction of feature points on 3D surfaces. Therefore we generalized the scale space method of Lindeberg [Lin98] to 2-manifolds in 3D and use the averaged mean curvature flow to build an analog representation over scales. We detect a salient point by checking if it is extremal both in the adjacent scales and in the adjacent mesh vertices. The transfer of the hessian condition has shown good results by thresholding unstable features. Furthermore, we have shown the robustness of our method by processing several example surfaces.

One problem of our approach is the dependency on the used flow. The mean curvature flow is not qualified to be used in a general application, because it tends to fragment specific objects. Because of this, we want to explore different flows and their properties, in order to find a more suitable one for our method.

Due to the fact that we use a scale space based detection, we obtain features, that are robust against noise on the surface. Only in the first scales wrong features were found.

In a matching application a descriptor could be used to additionally improve the descriptive power of our features. To get scale invariance, this descriptor could work with a radius proportional to the scale of its feature point.

In the future, we would like to modify our method to compute other types of features, as for example line features.

REFERENCES

[BKS+05] Benjamin Bustos, Daniel A. Keim, Dietmar Saupe, Tobias Schreck, and Dejan V. Vranić. Feature-based similarity search in 3d object databases. *ACM Comput. Surv.*, 37(4):345–387, 2005.

[CRT04] Ulrich Clarenz, Martin Rumpf, and Alexandru Telea. Robust feature detection and local classification for surfaces based on moment analysis. *IEEE Transactions on Visualization and Computer Graphics*, 10(5):516–524, 2004.

[GCO06] Ran Gal and Daniel Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph.*, 25(1):130–150, 2006.

[IJL+05] Natraj Iyer, Subramaniam Jayanti, Kuiyang Lou, Yagnanarayanan Kalyanaraman, and Karthik Ramani. Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design*, 37(5):509–530, 2005.

[JWBD86] Babaud J., A. P. Witkin, M. Baudin, and R. O. Duda. Uniqueness of the gaussian kernel for scale-space filtering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(1):26–33, 1986.

[Koe84] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[LG05] Xinju Li and Igor Guskov. Multiscale features for approximate alignment of point-based surfaces. In *Symposium on Geometry Processing*, pages 217–226, 2005.

[Lin98] Tony Lindeberg. Feature detection with automatic scale selection. *Int. J. Comput. Vision*, 30(2):79–116, 1998.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[MTS+05] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, 2005.

[NDK05] M. Novotni, P. Degener, and R. Klein. Correspondence generation and matching of 3d shape subparts. Technical report, University of Bonn, 2005.

[SF07] Philip Shilane and Thomas Funkhouser. Distinctive regions of 3d surfaces. *ACM Transactions on Graphics*, 26(2):Article 7, 2007.

[SOS04] Chen Shen, James F. O'Brien, and Jonathan R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. *ACM Trans. Graph.*, 23(3):896–904, 2004.

[TL07] Gary K. L. Tam and Rynson W. H. Lau. Deformable model retrieval based on topological and geometric signatures. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):470–482, 2007.

[TV04] Johan W. H. Tangelder and Remco C. Veltkamp. A survey of content based 3D shape retrieval methods. In *Proceedings of the Shape Modeling International*, pages 145–156, 2004.

[Wit83] Andrew P. Witkin. Scale-space filtering. In *8th Int. Joint Conference on Artificial Intelligence*, pages 1019–1022, 1983.

[WNK06] R. Wessel, M. Novotni, and R. Klein. Correspondences between salient points on 3d shapes. In *Vision, Modeling, and Visualization 2006 (VMV 2006)*, pages 365–372, 2006.

[Xu04] Guoliang Xu. Convergent discrete laplace-beltrami operators over triangular surfaces. In *Proceedings of the Geometric Modeling and Processing*, pages 195–204, 2004.

[ZF03] Barbara Zitová and Jan Flusser. Image registration methods: a survey. *Image Vision Computing*, 21(11):977–1000, 2003.

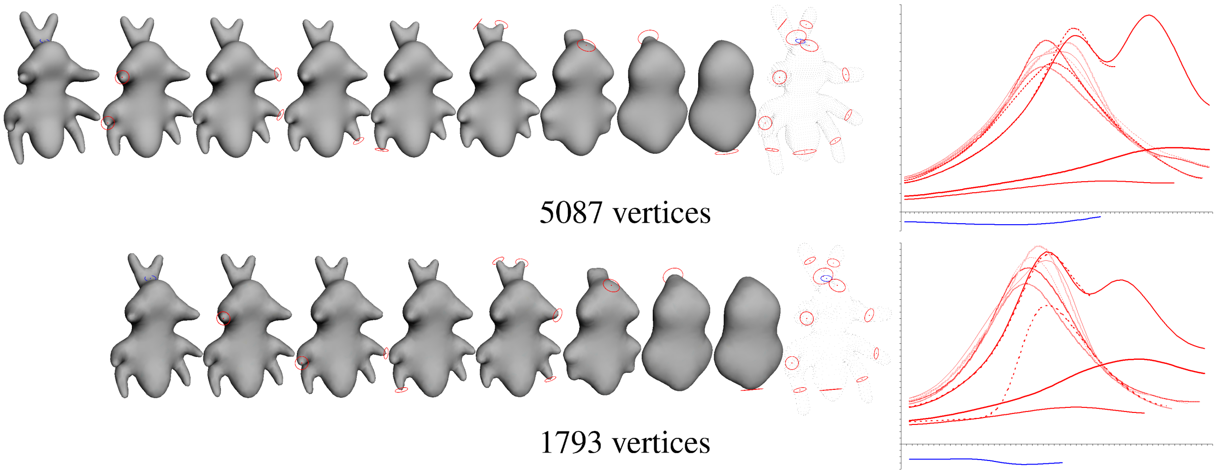


Figure 11: The feature points of an ant model with different sample rates. (Left) Smoothed models in scales of the features. (Right) Signatures.

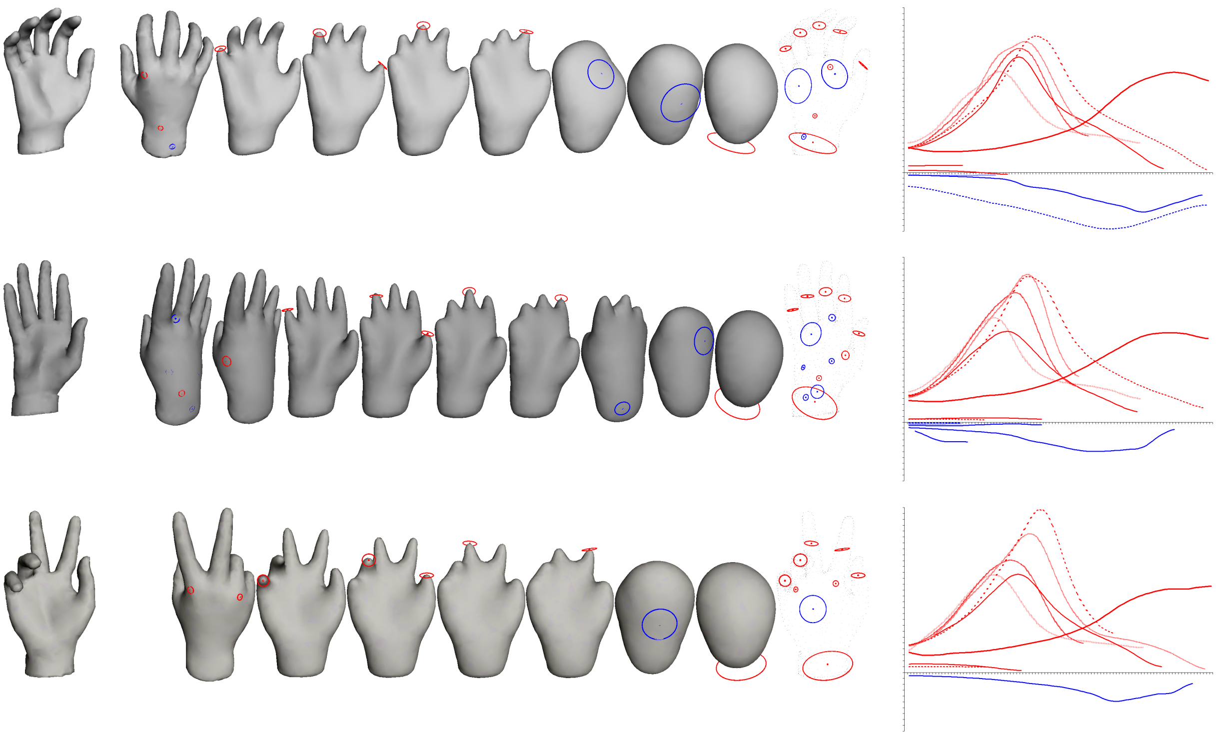


Figure 12: Feature points and signatures of three poses of a hand. (Left) Original models (approx. 1400 vertices). (Middle) Smoothed objects in scales of the features. (Right) Signatures.

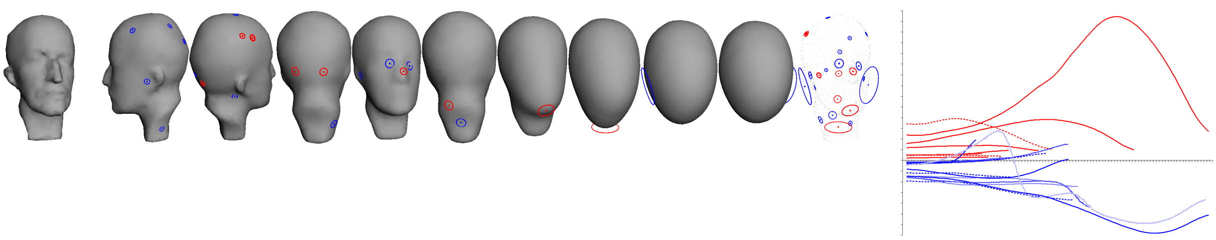


Figure 13: Feature points and signatures of the Max Planck model. (Left) Original model (approx. 1650 vertices). (Middle) Smoothed object in scales of the features. (Right) Signatures.