2013 IEEE Conference on Computer Vision and Pattern Recognition

Multi-Scale Curve Detection on Surfaces

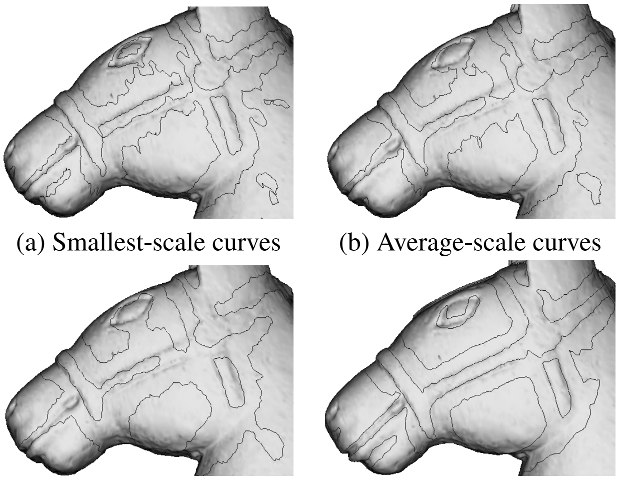
Michael Kolomenkin
Technion
michael.kolomenkin@gmail.com

Ilan Shimshoni
The University of Haifa
ishimshoni@is.haifa.ac.il

Ayellet Tal
Technion
ayellet@ee.technion.ac.il

Abstract

This paper extends to surfaces the multi-scale approach of edge detection on images. The common practice for detecting curves on surfaces requires the user to first select the scale of the features, apply an appropriate smoothing, and detect the edges on the smoothed surface. This approach suffers from two drawbacks. First, it relies on a hidden assumption that all the features on the surface are of the same scale. Second, manual user intervention is required. In this paper, we propose a general framework for automatically detecting the optimal scale for each point on the surface. We smooth the surface at each point according to this optimal scale and run the curve detection algorithm on the resulting surface. Our multi-scale algorithm solves the two disadvantages of the single-scale approach mentioned above. We demonstrate how to realize our approach on two commonly-used special cases: ridges & valleys and relief edges. In each case, the optimal scale is found in accordance with the mathematical definition of the curve.



(a) Smallest-scale curves (b) Average-scale curves

(c) Large-scale curve (d) Our multi-scale curves

Figure 1. **The benefit of using multi-scale curves.** When relief edges [10] are detected using a single scale, some features are missed and others are inaccurate (a)-(c). Conversely, when using multiple scales, the detected curves are more correct (d).

1. Introduction

3D feature curves on surfaces carry important information regarding the shape of the object. Therefore, a lot of effort has been devoted to charactering curves and detecting them. Examples of types of curves include ridges & valleys [19], parabolic curves [8], zero-mean curvature curves [8], demarcating curves [9], and relief edges [10], to name a few. Each type of curve is used to detect a different 3D feature. Curves on surfaces are equivalent to edges in images, which are basic low-level features in images. Consequently, 3D curves are inherently important in 3D shape analysis.

In images, each edge is associated with a scale. This scale is related to the image gradient; the steeper and stronger the edge, the smaller the scale. This is because steep edges are thinner, occupying a smaller area in the image than fuzzy moderate edges. This concept exists also in 3D curves on surfaces. For instance, the eye of the horse in Figure 1 has a smaller scale than its harness.

As illustrated in Figure 1, no single scale suffices to capture all the features. If the scale is too large, fine details are missed. On the other hand, if the scale is too small, coarse features are localized inaccurately and false features appear. However, most state-of-the-art curve detection algorithms use a single scale [1, 5, 9, 10, 19]. Moreover, the user is required to manually choose the "correct" scale.

Our goal is to propose a general framework for automatically estimating the optimal scale at each point on the surface. This general scheme can then be applied to every type of 3D curve, assuming it can be defined by the curvature and its derivatives. Hence, our technique not only eliminates the needed user intervention, but is also able to detect feaures of different scales on a single object.

A couple of algorithms address scale selection. Pauly et al. [20] propose a scheme that is designed for a single type of curves, defined as the loci of points whose curvature variation is persistent over all scales. It cannot be applied in a straightforward manner to other types of curves. Luo et al. [14] propose a method that is independent of the curve

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.36

type. Rather, they essentially apply multi-scale smoothing to the object, hopefully leaving the surface features intact. However, the scale must be proportional not only to the surface features, but also to the curve type. When the latter is not taken into account, the extracted curve might be inaccurate. Moreover, their approach is quite slow (might take hours), whereas ours takes only a couple of minutes.

Our technique is inspired by the scale-selection theory developed by Lindeberg [13] for images (on which SIFT is based). We extend this theory to three dimensions. Given a definition of a curve type, we show how to calculate the optimal scale directly from the definition. Briefly, every curve type is associated with a function,whose value indicates the strength of the feature. This function typically depends on the surface curvature and its derivatives. We show how to select a parameter, in order for this function to have a single maximum over the range of scales. We define the scale at which the maximum is obtained as the optimal scale. Using this local optimal scale, the surface is smoothed in a multiscale manner. On this surface, the curves are detected, utilizing the original curve detection algorithms.

We demonstrate the benefit of our approach by applying it to two popular types of curves: ridges & valleys and relief edges. We show that our curves outperform their counterparts computed with a manually-selected single scale, both in terms of accuracy and in terms of robustness to noise. This is especially evident in objects having features of several scales.

The paper is structured as follows. Section 2 provides the essential background. Section 3 formally defines the notion of optimal scale. Section 4 describes the algorithm for computing the optimal scale at every point. Section 5 applies the general method to two specific cases and demonstrates our results. We conclude in Section 6.

2. Background

This section describes the background on curve detection on surfaces and on multi-scale processing on surfaces.

Curves on surfaces: Curves on surfaces can be classified as view-dependent or view-independent. *View-dependent curves* depend not only on the differential geometric properties of the surface, but also on the viewing direction [1, 5, 7]. They change whenever the camera changes its position or orientation. *View-independent curves* depend solely on geometric properties of the surface [6, 8, 9, 10, 16, 19]. Our approach is general and applies to both categories of curves. For demonstration, we apply our approach to two types of view-independent curves. The first is *ridges & valleys* [19], which are the extrema of principal curvatures and the second are *relief edges* [9, 10], which are the loci of zero crossings of the curvature in the edge direction.

Mutli scale processing of surfaces: Multi-scale processing of surfaces can be divided into two separate, yet related, tasks. The first is the creation of *scale space*, which simultaneously represents the surface at different scales. The second is the *optimal scale selection*, which automatically selects the scale that best represents the surface locally.

1. Scale space representation: Scale space can be intuitively thought of as a collection of smoothed versions of the original surface. While in images, which are defined on a regular grid scale space, smoothing is a matter of consensus [13], for surfaces there are many ways to perform smoothing [3, 11, 17, 20, 21].

Formally, given a surface $S(u, v) : \mathbb{R}^2 \rightarrow \mathbb{R}^3$, its scale space representation is

$$S(u, v, t) : \mathbb{R}^2 \rightarrow \mathbb{R}^3,$$

where t is the scale parameter, which is proportional to the amount of smoothing applied to the object. In our work, we use the diffusion-based smoothing of [23], though other methods could also be used.

2. Optimal scale selection: In images, the optimal scale selection was developed for edge detection and point-based features by Lindeberg [13]. He proved that a function of the image spatial derivatives, which is normalized in a certain way, obtains a single maximum in scale space. The scale at which the maximum is obtained is termed the optimal scale.

Formally, let $g : \mathbb{R}^2 \rightarrow \mathbb{R}$ be an image, $L(\mathbf{x}; t)$ be its scale space representation, and $\delta_{\mathbf{x}^k}$ be its k^{th} derivative. Then, the γ-normalized function of the derivatives of the image, defined as

$$t^{\gamma/2} f(\delta_{\mathbf{x}^k} L(x; t)) \tag{1}$$

obtains a single maximum at the optimal scale t_o. The function f and the k^{th} derivative depend on the desired feature.

Optimal scale for surfaces was mostly used for interest point detection. At these points, a function of some surface properties, normalized by the scale parameter, obtains a maximum both in the spatial and in the scale domains. These properties can be either normals [4], curvatures [22], or functions of curvatures [18]. These SIFT-like features are applied together to photometric and geometric features in [3].

A different approach, to which our method belongs, explicitly chooses a region on the surface, where the variation of some surface property is small. For instance, [20] compute the ratio between the eigenvalues; [14] compute the area with the minimal descriptor length; [12] compute the optimal size of the support region for computing surface normals.

3. Definition of the Optimal Scale

Intuitively, the optimal scale at point $\mathbf{p}$ is the scale at which the likelihood that a curve passes through $\mathbf{p}$ is maximal. Therefore, we consider the likelihood to be proportional to the curve strength. However, the more smoothing applied to the surface, the weaker the curves become. To compensate for this, the strength is normalized by a function of the scale. Hence, the optimal scale is reformulated as the scale at which the normalized curve strength obtains a maximum in scale space.

Definition: Let curve c have an associated strength function $f(K)$, which depends on the curvature and the curvature derivatives K. Let t be the scale parameter, which is related to the amount of smoothing applied to the surface. Let $f(K(t))$ be the result of applying f to the smoothed surface. Then, the optimal scale s_c of curve c is defined as the scale at which the function obtains a maximum:

$$s_c = \arg\max_t t^\gamma f(K(t)), \tag{2}$$

where t^γ is the normalization coefficient, similar to what is used in Equation (1). We denote the expression $t^\gamma f(K(t))$ as the *normalized function*.

The strength function $f(K(t))$ monotonically decreases with scale, since as the surface becomes smoother, the features appear less prominent. Conversely, t^γ monotonically increases with scale. We should therefore choose γ correctly, so as to ensure a single maximum of the normalized function $t^\gamma f(K(t))$ in Equation (2). This single-maximum property is proved in Section 5 for two commonly-used spatial curve types: ridges & valleys and relief edges.

To get a feeling why this is true, recall that a surface can be presented locally, at every point on the surface, as a third degree polynomial (the *Monge form*) defined on the point's tangent plane. The surface curvature is proportional to the polynomial second derivative. According to Equation (2), the normalized second derivative obtains a single maximum in the scale space. Therefore, the surface curvature should also obtain a maximum in scale space.

Note that in featureless areas, the feature strength is zero for all scales. Thus, the optimal scale (the maximum) is undefined there. However, since there exist no curves in a featureless region and our goal is to detect curves, any scale chosen is valid and our definition holds.

4. Multi-Scale Curve Detection

The goal of this section is to describe an algorithm that detects the curves, given the definition of the optimal scale, described above. Our algorithm consists of three steps. First, we compute the optimal scale at every point on the surface. This computation depends on the type of curve we want to detect, and specifically on its corresponding strength function f. Second, we create a new surface, where each point is smoothed according its optimal scale. Obviously, this step does not depend on the type of curves. On this smoothed surface, the curves are finally computed using their original detection algorithms. This is in contrast to the regular curve detection algorithms, which perform a uniform smoothing on the whole surface, prior to detection.

Computing the optimal scale at every point: For points with features, we apply Equation (2) as is and obtain the optimal scale. For featureless points, any value of scale is valid, and yet we need to choose a single scale. Since our goal is to produce the optimally-scaled surface, the scale should be a continuous function over the surface. We therefore require that the following conditions hold:

1. The scale at a point with high strength is equal to the solution of Equation (2).

2. The scale changes smoothly along the surface.

Formally, the first condition is a simple equality constraint. The second condition is realized by requiring that the weighted Laplacian of the scalar scale function, defined on the surface, is zero. These two conditions are combined into a system of linear equations, as follows.

Let $\mathbf{p}$ be a point on the surface and $t_i(\mathbf{p})$ be the scale that solves Requirement 1. The final scale $t(\mathbf{p})$ is found by solving the following system:

$$\begin{aligned} t(\mathbf{p}) &= t_i(\mathbf{p}) \\ w(\mathbf{p})\Delta t(\mathbf{p}) &= 0, \end{aligned} \tag{3}$$

where $w(\mathbf{p})$ is a weight function inversely proportional to the strength at $\mathbf{p}$ and Δ is the Laplacian. We define $w(\mathbf{p})$ as $1/(1+af)$, where f is the strength function value at the point and a is chosen so that $af_{\max} = 9$.

In our work, we assume that the surface is represented by a triangular mesh. The Laplacian Δ of a scalar function t at point $\mathbf{p}$ on a mesh is calculated as in [15]:

$$\Delta t(\mathbf{p}) = \frac{1}{2A} \sum_{j \in N(\mathbf{p})} (\cot(\gamma_j) + \cot(\delta_j))(t(\mathbf{p}) - t(\mathbf{p}_j)), \tag{4}$$

where $N(\mathbf{p})$ is the set of neighbors of $\mathbf{p}$ on the mesh, A is the Voronoi area of $\mathbf{p}$, and γ_j and δ_j are the angles opposite to $\mathbf{p}\mathbf{p}_j$ (see Figure 2). We solve the system of equations (3) with an SVD-based solver [2].

Creating an optimally-smoothed surface: After the optimal scale at each point has been computed, we create a surface in which each point is smoothed to its optimal scale. This is done as follows. First, the surface is smoothed uniformly for each scale, using [23]. For each of these surfaces, we maintain the locations of its vertices. Next, we create a surface for which the coordinates of every vertex are taken from the surface of its corresponding scale. Finally, the resulting surface is smoothed in order to remove artifacts.

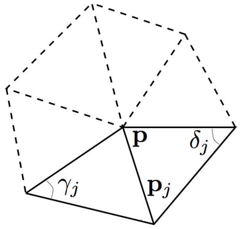


Figure 2. **Notations of Equation** (4)**.** The Laplacian Δ of a scalar function t at point $\mathbf{p}$ on a mesh is a linear combination of the values of t on the neighbors $\mathbf{p}_j$ of $\mathbf{p}$.

5. Specific Cases

In this section we demonstrate how to apply our method to commonly-used curves: ridges & valleys and relief edges. We show how to choose the strength function f and the normalization coefficient γ from Equation (2), which are used in the first step of our algorithm (Section 4).

Recall that the normalization coefficient γ should be chosen such that to ensure a single maximum of the normalized strength function. In order to simplify the task, we make three assumptions. First, we make the standard assumption that the surface in the curve's neighborhood is a function defined on the tangent plane, i.e. the surface is defined by $s(x, y)$. Second, we assume that $s(x, y)$ is constant along the curve, i.e. the surface can be modeled locally by a 1D function. For example, if the curve has direction y, then $s(x, y) \equiv s(x)$. In other words, we assume that $s(x, y)$ is locally developable in the vicinity of the curve. Third, we make the commonly-used assumption that the smoothing process can be approximated by convolving the function with a Gaussian. Our experiments show that this approximation is sufficiently accurate.

Having made these assumptions, we perform the following general steps for finding γ. For each curve type we need to realize these steps differently.

1. Choose the strength function.
2. Choose a 1D function $s(x)$ that represents the surface locally.
3. Derive the expression for the scale at which the normalized strength obtains a maximum. This maximum is the optimal scale.
4. Choose γ values for which the expression in step 3 is maximized by a single scale value.

5.1. Ridges & valleys

The most prevalent curves on surfaces are ridges & valleys [19]. They are similar to their geographical counterparts and usually indicate sharp changes in the surface orientation. A ridge (valley) point is a point on a manifold, where the positive (negative) principal curvature obtains a maximum (minimum) along its principal direction. We now discuss the four steps mentioned above, for the case of ridges, whereas the case of valleys is similar.

Strength function f: The strength function f we use is the maximal value of the curvature. This is the standard method to measure the strength of ridges.

Surface representation: We approximate a ridge $s(x; t_0)$ of scale t_0 with a Gaussian of standard deviation $\sigma = \sqrt{t_0}$:

$$s(x; t_0) = \frac{1}{\sqrt{2\pi t_0}} e^{-x^2/2t_0}.$$

The ridge point is obtained at $x = 0$.

We assume that the scale space is represented by convolutions with Gaussians with smoothing parameter t:

$$s(x; t + t_0) = s(x; t_0) * g(x; t). \tag{5}$$

Thus, the ridge $s(x; t_0)$ at scale t is

$$s(x; t + t_0) = \frac{1}{\sqrt{2\pi(t_0 + t)}} e^{-x^2/2(t_0+t)}.$$

Optimal scale: We want to show that the normalized curvature of $s(x; t_0)$ obtains a single maximum in scale space at $x = 0$. First, we express the normalized curvature at $x = 0$ as a function of t and γ:

$$t^\gamma \kappa(x; t_0 + t). \tag{6}$$

Then, we take its derivative with respect to t and prove that this derivative is equal to zero only at a single t. Let us start with finding the expression for $\kappa(x; t_0 + t)$. The curvature $\kappa(x)$ of a curve $s(x)$ is known to be:

$$\kappa(x) = -\frac{s''(x)}{(1 + s'(x)^2)^{3/2}}, \tag{7}$$

where the derivatives are with respect to x.

We need to compute the derivatives of the surface (curve) $s'(x)$ and $s''(x)$. It is easy to see that the first derivative is:

$$s'(x; t_0 + t) = \frac{-x}{t_0 + t} \cdot \frac{e^{-x^2/2(t_0+t)}}{\sqrt{2\pi(t_0 + t)}}$$

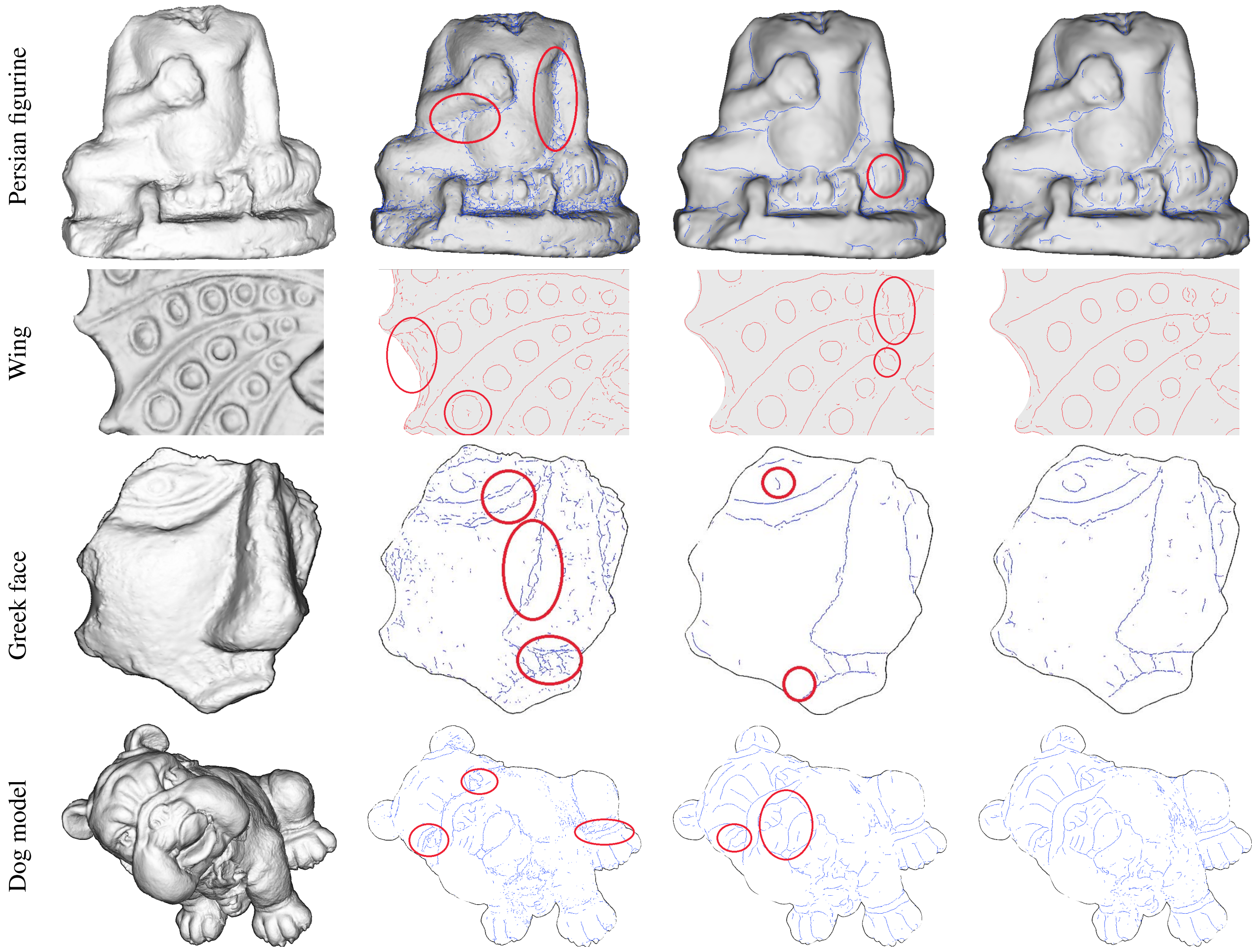
and the second derivative is:

$$s''(x; t_0 + t) = \left(\frac{-1}{t_0 + t} + \frac{x^2}{(t_0 + t)^2} \right) \cdot \frac{e^{-x^2/2(t_0+t)}}{\sqrt{2\pi(t_0 + t)}}.$$

Substituting $x = 0$, we obtain

$$s'(0; t_0 + t) = 0, \tag{8}$$

$$s''(x; t_0 + t) = \frac{-1}{t_0 + t} \cdot \frac{1}{\sqrt{2\pi(t_0 + t)}} = \frac{-1}{\sqrt{2\pi}(t_0 + t)^{3/2}}. \tag{9}$$



(a) surface (b) small scale (c) large scale (d) our multiscale

Figure 3. **Ridges and valleys on objects with various scales.** When the chosen scale is small (b), the results are noisy. When the scale is large (c), some features disappear. With our multi-scale approach (d), all the features are nicely captured and the curves are smooth. The red circles indicate problematic areas.

Therefore, by substituting Equations (8)-(9) into Equations (7)-(6), we get that the value of the strength function at $x = 0$ for different scales t and different normalization coefficients γ is:

$$t^{\gamma}\kappa(x; t_0 + t)\Big|_{x=0} = \frac{t^{\gamma}}{\sqrt{2\pi}(t_0 + t)^{3/2}}. \qquad (10)$$

In order to check when Equation (10) obtains a maximum, we set its derivative with respect to t to 0:

$$\frac{d}{dt}t^{\gamma}\kappa(x; t_0 + t)\Big|_{x=0} = 0, \quad \text{or}$$

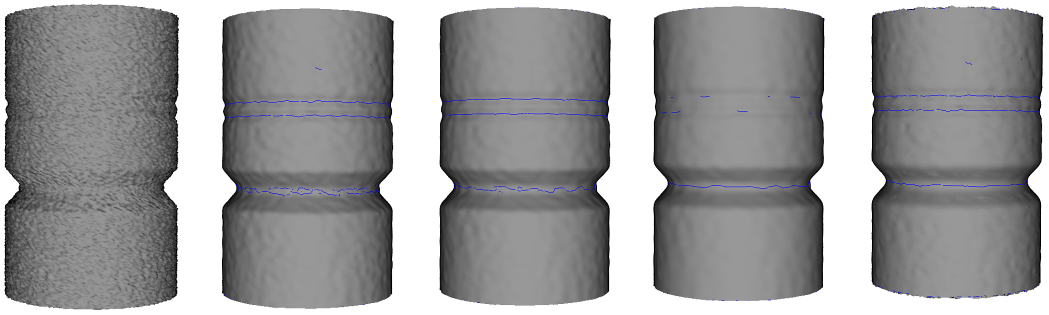
$$\frac{\gamma t^{\gamma-1}}{\sqrt{2\pi}(t_0 + t)^{3/2}} - \frac{3t^{\gamma}}{2\sqrt{2\pi}(t_0 + t)^{5/2}} = 0.$$

After some simple algebraic manipulations, we obtain an equation for the optimal scale:

$$t_{\text{max}} = 2t_0/(3 - 2\gamma). \qquad (11)$$

Choice of γ: We conclude that when $\gamma < 1.5$, the optimal scale is unique. In our experiments, we use $\gamma = 1$. We observed that small changes of γ do not influence the results.

Results: Figure 3 shows ridges and valleys on objects consisting of features of different scales. When the chosen scale is small (b), the results are noisy. When we increase the scale (c), the curves become smoother, but some features disappear. We were unable to find a single scale that yields a good trade-off between smoothness and detectability. Conversely, with our multi-scale approach (d), the curves are smooth and even small details are nicely captured. See the



(a) Input (b) fine scale (c) coarse scale (d) coarsest scale (e) multi scale

Accuracy: 85% 77% 42% 91%

Figure 4. **Quantitative evaluation.** The noisy cylinder has both fine and coarse valleys (a). The results of our multi-scale approach (e) outperforms those of the single-scale approach (b-d). The accuracy measure is given underneath the images.

supplementary for additional results.

To provide quantitative evaluation, we created a synthetic example, for which we can calculate the ground truth. It is a cylinder with ridges and valleys of a couple of different scales, to which we added noise, as shown in Figure 4. We ran the single-scale algorithm using various single scales and our multi-scale algorithm. When the (single) scale is small, the coarse features are not detected accurately. When it is large, the fine features disappear. With our multi-scale approach, the features are nicely captured. In terms of error, we computed the percentage of the ground-truth valleys for which there exist real valleys within a predefined distance. In practice, we counted the number of faces with ground truth valleys for which there exists a face with a detected valley within 0.5% of the cylinder radius. While the accuracy of the multi-scale results is 91%, for the single scale algorithm the results are between 79% for the finest scale to 42% for the coarsest, The results are: Scale 0 (finest): 79%, Scale 1: 85%, Scale 2: 77%, Scale 3 (coarsest): 42%, and Multiscale: 91%.

5.2. Relief edges

Relief edges are defined as zero crossings of the curvature in the direction of the step edge model that best approximates the surface locally [10]. They run on the slopes between ridges and valleys and are parallel to them.

Strength function f: As a strength function f, we employ the curvature derivative with respect to the arclength λ in the edge direction, as proposed in [10]. Thus,

$$f = \frac{\partial \kappa(x;t_0)}{\partial \lambda}.$$

Surface representation: A relief edge is represented, by definition, by a smoothed step edge. Let $s(x;t_0)$ be a relief edge of an initial scale t_0 (an ideal step edge smoothed with a Gaussian of standard deviation $\sigma_0 = \sqrt{t_0}$):

$$s(x;t_0) = \frac{1}{\sqrt{2\pi t_0}} \int_{-\infty}^{x} e^{-u^2/2t_0} du.$$

Optimal scale: The proof is similar to that in Section 5.1. We want to show that the normalized curvature derivative of $s(x;t_0)$ obtains a single maximum in scale space at $x = 0$. First, we express the normalized curvature at $x = 0$ as a function of t and γ:

$$t^\gamma f = t^\gamma \frac{\partial \kappa(x;t+t_0)}{\partial \lambda}. \tag{12}$$

Then, we take its derivative with respect to t and prove that it is equal to zero only at a single t. The curvature derivative is:

$$\frac{\partial \kappa}{\partial \lambda} = \frac{\partial \kappa}{\partial x} \cdot \frac{\partial x}{\partial \lambda}. \tag{13}$$

We now show how to compute $\partial k/\partial x$ and $\partial k/\partial \lambda$ and then combine them. To compute $\partial k/\partial x$, we take the derivative of the curvature defined in Equation (7). The scale space is generated using a convolution with a Gaussian (Equation (5)):

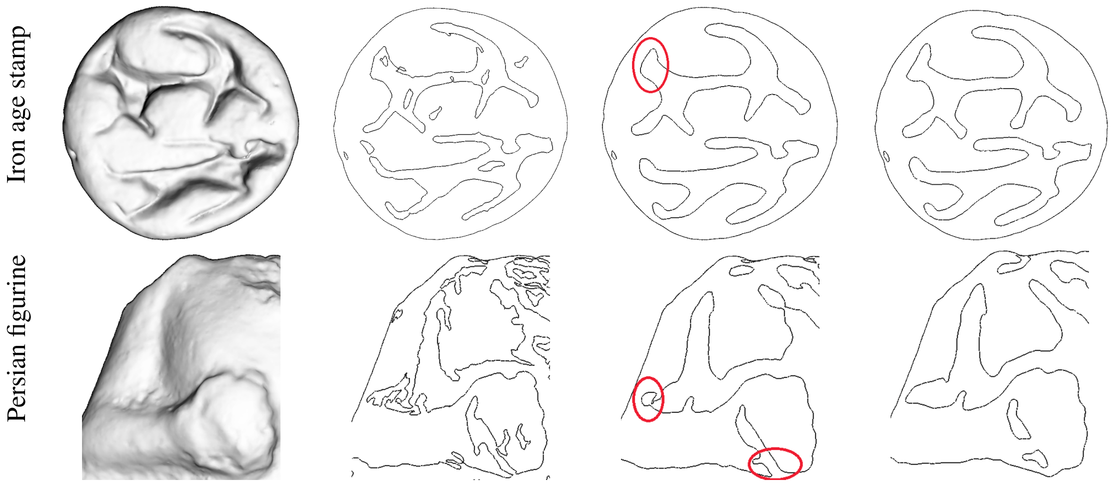
$$s(x;t+t_0) = \frac{1}{\sqrt{2\pi(t_0+t)}} \int_{-\infty}^{x} e^{-u^2/2t_0+t} du.$$

Then, the derivative of $s(x)$ with respect to x is

$$s'(x;t_0+t) = \frac{e^{-x^2/2(t_0+t)}}{\sqrt{2\pi(t_0+t)}}$$

and the second derivative is

$$s''(x;t_0+t) = \frac{-x}{t_0+t} \cdot \frac{e^{-x^2/2(t_0+t)}}{\sqrt{2\pi(t_0+t)}}.$$



(a) surface (b) small scale (c) large scale (d) our multi-scale

Figure 5. **Relief edges on objects having various scales.** When the scale is small (b), the resulting single scale curves are noisy. When the scale is big (c), the single scale curves are smooth but not accurate enough. For example, they do not capture the head of the dog in the top row and create topological mistakes in the arm on the bottom row. Our multi-scale relief edges are more accurate.

We now proceed to compute the curvature $\kappa(x)$

$$\kappa(x) = \frac{-xe^{-x^2/2(t_0+t)}}{(t_0+t)^{3/2}\sqrt{2\pi}} \cdot \frac{1}{[1+e^{-x^2/(t_0+t)}/(2\pi(t_0+t))]^{3/2}}$$

and the curvature derivative

$$\begin{aligned} \left.\frac{\partial k}{\partial x}\right|_{x=0} &= \frac{-e^{-\frac{x^2}{2(t_0+t)}}}{(t_0+t)\sqrt{2\pi(t_0+t)}} \cdot \left.\frac{1}{\left[1+\frac{e^{-\frac{x^2}{t_0+t}}}{2\pi(t_0+t)}\right]^{3/2}}\right|_{x=0} \\ &= \frac{2\pi}{\left(1+2\pi(t_0+t)\right)^{3/2}}. \end{aligned}$$

After having computed $\partial\kappa/\partial x$ of Equation (13), we now derive $\partial x/\partial\lambda$:

$$\partial\lambda = \sqrt{\partial x^2 + \partial y^2}, \frac{\partial\lambda}{\partial x} = \sqrt{1+s'(x)^2},$$

$$\frac{\partial x}{\partial\lambda} = \frac{1}{\sqrt{1+s'(x)^2}}.$$

Equation (13) now becomes

$$\frac{\partial\kappa}{\partial\lambda} = \frac{2\pi}{\left(1+2\pi(t_0+t)\right)^{3/2}} \cdot \frac{1}{\sqrt{1+s'(x)^2}} =$$

$$\frac{2\pi}{\left(1+2\pi(t_0+t)\right)^{3/2}} \cdot \frac{\sqrt{2\pi(t_0+t)}}{\sqrt{1+2\pi(t_0+t)}} = \frac{4\pi^2\sqrt{t_0+t}}{(1+2\pi(t_0+t))^2}.$$

In turn, Equation (12) now becomes

$$t^\gamma \frac{\partial\kappa(x;t+t_0)}{\partial\lambda} = t^\gamma \frac{4\pi^2\sqrt{t+t_0}}{(1+2\pi(t+t_0))^2}. \tag{14}$$

Equation (14) specifies the value of the strength function at $x = 0$ for different scales t and different normalization coefficients γ. We next find when it obtains a maximum. Taking the derivative with respect to t and setting it to zero, we obtain a quadratic equation whose roots are:

$$t_{\max} = (A \pm B)/C, \tag{15}$$

where

$$\begin{aligned} A &= 2\gamma - 6\pi t_0 + 8\pi\gamma t_0 + 1, \\ B &= (4\gamma^2 + 40\pi\gamma t_0 + 4\gamma + 36\pi^2 t_0^2 - 12\pi t_0 + 1)^{1/2}, \\ C &= 4(3\pi - 2\pi\gamma). \end{aligned}$$

The derivation of Equation (15) is given in Appendix A.

Choice of γ: We show in Appendix A that $(A - B)/C$ is always negative and thus irrelevant, whereas $(A + B)/C$ is always positive for $\gamma < 1.3$. Hence, for $\gamma < 1.3$ the optimal scale is unique.

Results: Figure 5 depicts relief edges on surfaces having features of various scales. It compares two single-scale relief edges with our multi-scale edges. The results of the large scale were found to be the best over all scales. Even though the large single-scale result of the Iron Age stamp is pretty, the head is portrayed inaccurately. This is so, since the depth of the relief varies, thus a single scale is insufficient. On the other hand, our multi-scale approach detects the edges accurately. On the bottom row, our multi-scale curves are much smoother than the best single-scale result, when tested on the arm of the figurine from Figure 3. To recap, our results are more accurate and smoother than those produced by the single-scale approach.

Additional benefit of our approach: In addition to the superiority of our approach in cases where the objects have a variety of scales, it is also beneficial in the case when objects have a single scale. While in all the single-scale approaches, the user needs to manually select the scale parameter, in our approach no manual tuning is necessary.

6. Conclusion

This paper presented a framework for automatic estimation of the optimal scale for curve detection on surfaces. It can be applied to any curve type, as long as the curve has a strength function based on the curvature and its derivatives. This requirement is satisfied by most curve types.

Our experiments show that on objects composed of features of various scales, the curves obtained by our method outperform those computed by the original single-scale algorithms. On objects that consist primarily of features of a single scale, the benefit of our algorithm is that it does not require manual parameter tuning.

Acknowledgements: This research was supported in part by the Israel Science Foundation (ISF) 1420/12 and the Ollendorff Foundation.

References

[1] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella. Suggestive contours for conveying shape. *TOG*, 22(3):848–855, 2003. 1, 2

[2] B. Flannery, W. Press, S. Teukolsky, and W. Vetterling. *Numerical recipes in C*. Cambridge Univ Press, 1992. 3

[3] T. Hou and H. Qin. Efficient computation of scale-space features for deformable shape correspondences. In *ECCV*, pages 384–397, 2010. 2

[4] J. Hua, Z. Lai, M. Dong, X. Gu, and H. Qin. Geodesic distance-weighted shape vector image diffusion. *TVCG*, 14(6):1643–1650, 2008. 2

[5] T. Judd, F. Durand, and E. Adelson. Apparent ridges for line drawing. *TOG*, 26(3):19:1–7, 2007. 1, 2

[6] D. Katsoulas and A. Werber. Edge detection in range images of piled box-like objects. In *ICPR*, pages 2:80–84, 2004. 2

[7] J. Koenderink. What does the occluding contour tell us about solid shape. *Perception*, 13(3):321–330, 1984. 2

[8] J. Koenderink. *Solid shape*. Cambridge Univ Press, 1990. 1, 2

[9] M. Kolomenkin, I. Shimshoni, and A. Tal. Demarcating curves for shape illustration. *TOG*, 27(5):157:1–9, 2008. 1, 2

[10] M. Kolomenkin, I. Shimshoni, and A. Tal. On edge detection on surfaces. In *CVPR*, pages 2767–2774, 2009. 1, 2, 6

[11] M. Kolomenkin, I. Shimshoni, and A. Tal. Prominent field for shape processing of archaeological artifacts. *IJCV*, 94(1):89–100, 2011. 2

[12] J. Lalonde, R. Unnikrishnan, N. Vandapel, and M. Hebert. Scale selection for classification of point-sampled 3D surfaces. In *3DIM*, pages 285–292, 2005. 2

[13] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *IJCV*, 30(2):117–154, 1998. 2

[14] T. Luo, R. Li, and H. Zha. 3D line drawing for archaeological illustration. *IJCV*, 94(1):23–35, 2011. 1, 2

[15] M. Meyer, M. Desbrun, P. Schroder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. *VisMath*, 3(7):34–57, 2002. 3

[16] O. Monga, R. Deriche, G. Malandain, and J. Cocquerez. Recursive filtering and edge tracking: two primary tools for 3D edge detection. *IVC*, 9(4):203–214, 1991. 2

[17] J. Novatnack and K. Nishino. Scale-dependent 3D geometric features. In *ICCV*, pages 1–8, 2007. 2

[18] J. Novatnack and K. Nishino. Scale-dependent/invariant local 3D shape descriptors for fully automatic registration of multiple sets of range images. In *ECCV*, pages 440–453, 2008. 2

[19] Y. Ohtake, A. Belyaev, and H. Seidel. Ridge-valley lines on meshes via implicit surface fitting. *TOG*, 23(3):609–612, 2004. 1, 2, 4

[20] M. Pauly, R. Keiser, and M. Gross. Multi-scale feature extraction on point-sampled surfaces. *CGF*, 22(3):281–289, 2003. 1, 2

[21] M. Pauly, L. Kobbelt, and M. Gross. Point-based multiscale surface representation. *TOG*, 25(2):177–193, 2006. 2

[22] M. Reuter, F. Wolter, and N. Peinecke. Laplace–Beltrami spectra as Shape-DNA of surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006. 2

[23] G. Taubin. Curve and surface smoothing without shrinkage. In *ICCV*, pages 852–860, 1995. 2, 3

A. Maxima of normalized curvature derivative of a step edge

Here, we find the values of t at which Equation (14) obtains maximum. These are the values at which the derivative of Equation (14) is equal to zero.

$$\frac{d}{dt}\frac{t^{\gamma}\sqrt{t+t_0}}{(1+2\pi(t+t_0))^2}=0$$

$$\frac{\gamma t^{\gamma-1}(t+t_0)^{1/2}}{(1+2\pi(t+t_0))^2}+\frac{t^{\gamma}}{2(t+t_0)^{1/2}[(1+2\pi(t+t_0))^2]}-$$

$$\frac{2\pi 2t^{\gamma}(t+t_0)^{1/2}}{[(1+2\pi(t+t_0))^3]}=0.$$

After several simple manipulations:

$$\frac{\gamma(t+t_0)}{1}+\frac{t}{2}-\frac{2\pi 2t(t+t_0)}{[(1+2\pi(t+t_0))]}=0, \quad \text{or}$$

$$2\pi(2\gamma-3)t^2+(2\gamma-6\pi t_0+8\pi\gamma t_0+1)t+2\gamma t_0(2\pi t_0+1)=0.$$

This is a quadratic equation whose roots t_i are

$$t_i=[2\gamma-6\pi t_0\pm(4\gamma^2+40\pi\gamma t_0+4\gamma+36\pi^2 t_0^2-12\pi t_0+1)^{1/2}+$$

$$8\pi\gamma t_0+1]/(4(3\pi-2\pi\gamma)).$$

A positive solution to this equation always exists for $0<\gamma<1.3$.