Height Ridge Computation and Filtering for Visualization

Ronald Peikert[*]   Filip Sadlo[†]

ETH Zurich

Abstract

Motivated by the growing interest in the use of ridges in scientific visualization, we analyze the two height ridge definitions by Eberly and Lindeberg. We propose a raw feature definition leading to a superset of the ridge points as obtained by these two definitions. The set of raw feature points has the correct dimensionality, and it can be narrowed down to either Eberly's or Lindeberg's ridges by using Boolean filters which we formulate. While the straight-forward computation of height ridges requires explicit eigenvalue calculation, this can be avoided by using an equivalent definition of the raw feature set, for which we give a derivation. We describe efficient algorithms for two special cases, height ridges of dimension one and of co-dimension one. As an alternative to the aforementioned filters, we propose a new criterion for filtering raw features based on the distance between contours which generally makes better decisions, as we demonstrate on a few synthetic fields, a topographical dataset, and a fluid flow simulation dataset. The same set of test data shows that it is unavoidable to use further filters to eliminate false positives. For this purpose, we use the angle between feature tangent and slope line as a quality measure and, based on this, formalize a previously published filter.

Index Terms: I.3.8 [Computer Graphics]: Applications; I.4.6 [Image Processing and Computer Vision]: Segmentation—Edge and feature detection; J.2 [Physical Sciences and Engineering]: Engineering

1 Introduction

Ridges and valleys are the main structures of a surface topography, and their mathematical idealization as curves was an objective already in the mid-nineteenth century work by de Saint-Venant[1] and Breton de Champ [3]. Ridges and valleys are still of importance for geomorphology [25], but in addition, they have proved to be useful in many other fields. In image analysis and computer vision, a digital image can be seen as a sampled scalar field or *height field*, making ridges available as characteristic structures [9, 15, 23, 17], complementary to the more popular edges. Various existing definitions of ridges can be extended to volumetric data, making ridges an important tool also for scientific visualization [6]. Besides medical visualization, ridges have become popular in flow visualization, e.g. as indications of vortex cores [26] or flow separation [24]. Recently, also two-dimensional ridges, i.e. ridge surfaces, in volumetric data were used by Kindlmann et al. [13] for visualization of diffusion tensor MRI data, by Sahner et al. [22] for visualization of vorticity and strain, and by Sadlo et al. [21] for separating regions of different flow behavior in unsteady vector fields.

A concept of ridges and valleys exists not only for height fields but also for surfaces in space. This type of ridges, of which the geodesic or maximum curvature ridge [20] is the one best known, is used e.g. in non-photorealistic rendering for enhancing salient features of a surface [11, 2]. At first glance the two problems of finding ridges in height fields and on surfaces seem identical, at least in the case of a 2D height field where the graph of the field is a surface. However, for a geodesic ridge, an important property is rotation invariance, i.e. it moves with the surface when that is rotated. In contrast to this, a ridge in a height field is required to be invariant not with respect to rotation in space but to scaling of the height.

There are two fundamentally different approaches to the definition of a ridge in a height field. A well known concept is the watershed, and its counterpart, the watercourse. It is obtained by locating all saddle points and by integrating from these along the positive and negative gradient. In the terminology of vector field topology [10], watersheds and watercourses constitute the topological skeleton. The second approach is to define ridges and valleys not as integral curves but as solution manifolds of algebraic equations which involve the height field and its derivatives. The most prominent ridge definition of this type is the height ridge [5]. There have been disputes on whether watersheds or height ridges are the "correct" ridges. Koenderink and van Doorn used an example, which they called the "curved gutter" [14], as an argument against height ridges. However, as we show in Appendix A, similar examples can be given that favor height ridges over watersheds.

The two types of methods yield different results in general and should be understood as complementary approaches having both their strengths and weaknesses. One the one hand, methods based on point-wise information are limited by their inherently "narrow view". But on the other hand, methods based on integration have the problems of sensitivity to small perturbations and of remote effects. In the case of watersheds this means for instance that any monotonically ascending ridge is not recognized as a watershed, but if it is almost horizontal at some point, then a small perturbation suffices to create a saddle point that causes the ridge to become a watershed. This is illustrated by an example in Appendix B. It might be interesting to note that already the famous mathematician Camille Jordan pointed out [12] that there is in general no slope line (i.e. integral curve of the gradient) that is characterized by special properties on its entire length, and that therefore only segments of slope lines can be attributed to be a ridge or a valley.

A very similar situation exists for the problem of vortex axis detectors in 3D vector fields. On the one hand there is the topologically based definition of vortex axes as streamlines emanating from critical points of saddle focus type. And on the other hand, there are local methods based on derivatives [27, 18].

There exist techniques that can bridge the gap between local and global methods. One such class are scale-space techniques, which extend the notion of "local" to a neighborhood of a size that grows with the observation scale. A second approach uses finite-time instead of infinite-time integration that would be required for topological features such as watersheds [8]. We will in in this paper restrict ourselves to the study of locally detectable features, with the goal of analyzing variants of the height ridge definition for their applicability in visualization.

[*] e-mail: peikert@inf.ethz.ch

[†] e-mail:sadlo@inf.ethz.ch

2 HEIGHT RIDGES

2.1 Height Ridge Definition

The concept of a height ridge dates back to the 19th century work of de Saint-Venant [1]. It has been formalized by Haralick [9] and elegantly reformulated by Eberly [5] and Lindeberg [15] for the general case of k-dimensional ridges in $\mathbb{R}^n$.

A *(maximum convexity) height ridge* of dimension k in a scalar field $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is defined as the set of points where

$$\frac{\partial f}{\partial y_1} = \cdots = \frac{\partial f}{\partial y_{n-k}} = 0 \tag{1}$$

$$\frac{\partial^2 f}{\partial y_1^2}, \cdots, \frac{\partial^2 f}{\partial y_{n-k}^2} < 0 \tag{2}$$

in a local coordinate frame $y_1, \cdots, y_n$ defined by the eigenvectors of the Hessian matrix $\mathbf{H} = \frac{\partial^2 f}{\partial x_i \partial x_j}$. Such a set of n orthogonal eigenvectors can be computed even in the case of multiple eigenvalues. The ordering of the eigenvalues $\lambda_i = \frac{\partial^2 f}{\partial y_i^2}$ is defined as follows.

Eberly's version:

$$\lambda_1 \leq \cdots \leq \lambda_n \tag{3}$$

Lindeberg's version:

$$|\lambda_1| \geq \cdots \geq |\lambda_n| \tag{4}$$

The counterpart of the height ridge, a "height valley" is simply obtained as a height ridge of the negative field $-f$. It is therefore not necessary to explicitly deal with the case of valleys, so we will in the sequel mostly just speak of ridges.

2.2 Raw Features and Filters

To allow for a common treatment of the two different definitions, we separate the process of solving equations (1) from the verification of the inequalities (2–4). We call a point a *raw feature point* if it solves (1) for an unspecified ordering of the eigenvectors. A raw feature point has the property that f has a horizontal tangent in at least $n-k$ of the n directions given by the local eigenvector frame. The set of raw feature points, which we denote by $\mathscr{R}_k$, is a superset of the height ridges (and valleys) in the sense of Eberly's definition. $\mathscr{R}_k$ can be reduced to that set by applying a point-wise Boolean filter (or predicate) $F_E(\mathbf{x}, f)$ which tests for (1) and (2) after reordering the eigenvalues by (3). In words this means, that the point must be a maximum in at least $n-k$ of the eigenvector directions and in the remaining ones there is either a weaker downward bend or an (unrestricted) upward bend. Analogously, the filter F_L based on Lindeberg's definition uses the ordering (4). Any point accepted by F_L is also accepted by F_E, but not vice versa, because F_L requires that a possible upward bend is also weaker than the downward bend in the first $n-k$ eigenvector directions. The difference between the two height ridge definitions is discussed in Majer's dissertation [17].

The effect of these Boolean filters is a removal of curve segments which are neither ridges nor valleys. Damon [4] called such segments *connector curves*. In practice, additional filters can be used for further refining the set of feature points. Typically, one wants to remove features that are below a certain minimal "feature strength". In the case of height ridges, such a feature strength could be defined e.g. by the absolute second derivative across the ridge. Another reason for adding more filter stages is that both F_E and F_L still accept points that must be considered as false positives.

2.3 Equivalent Characterization of Raw Height Ridges

For the set $\mathscr{R}_k$ of raw feature points the following equivalence holds:

A point $\mathbf{x}$ belongs to $\mathscr{R}_k$ if and only if at $\mathbf{x}$ the $k+1$ vectors $\mathbf{g}, \mathbf{Hg}, \cdots, \mathbf{H}^k\mathbf{g}$ are linearly dependent, that is,

$$\operatorname{rank}\left(\mathbf{H}^0\mathbf{g} | \cdots | \mathbf{H}^k\mathbf{g}\right) \leq k. \tag{5}$$

For the proof, let us first assume that (1) holds. We can write the gradient $\mathbf{g}$ in eigenvector basis as

$$\mathbf{g} = \sum_{i=1}^{n} a_i \mathbf{y}_i. \tag{6}$$

Multiplying $\mathbf{g}$ with j^{th} powers of $\mathbf{H}$ yields

$$\mathbf{H}^j\mathbf{g} = \sum_{i=1}^{n} \lambda_i^j a_i \mathbf{y}_i. \tag{7}$$

Since, by (1), $\mathbf{g}$ lies in a subspace spanned by k eigenvectors, $n-k$ of the coefficients, say $a_{k+1}, \cdots, a_n$, are zero. Consequently, the matrix

$$\left(\mathbf{H}^0\mathbf{g} | \cdots | \mathbf{H}^k\mathbf{g}\right) = \begin{pmatrix} \lambda_1^0 a_1 & \cdots & \lambda_1^k a_1 \\ \cdots & \cdots & \cdots \\ \lambda_n^0 a_n & \cdots & \lambda_n^k a_n \end{pmatrix} \tag{8}$$

has at most k nonzero rows, and hence a rank of at most k.

Conversely, assume that (5) holds. The eigenvector basis can now be chosen such that for each multiple eigenvalue, there is a single $\mathbf{y}_i$ for which $\mathbf{g}$ has a nonzero coefficient a_i. Then the matrix (8) can have at most k nonzero rows, because otherwise (assuming again that the nonzero rows are at the top and the zero rows at the bottom), it has an upper $k+1$ by $k+1$ submatrix which is a Vandermonde matrix with rows scaled by nonzero constants. This submatrix would then be regular and thus have a rank of $k+1$ contradicting our assumption. Hence, $\mathbf{g} = \nabla f$ lies in the subspace spanned by the first k eigenvectors of $\mathbf{H}$, which implies (1).

The equivalence of (1) and (5) can be used for a more efficient computation of $\mathscr{R}_k$. The straight-forward computation requires explicit calculation of eigenvalues which is a time-consuming operation, especially in higher dimensions. The alternative, evaluating (5), only involves matrix-vector multiplications and determinants. But, of course, points of $\mathscr{R}_k$ cannot practically be found just by sampling. Therefore, either of the two approaches also requires a method for finding such points. For a few special cases of k and n, we will present such methods below.

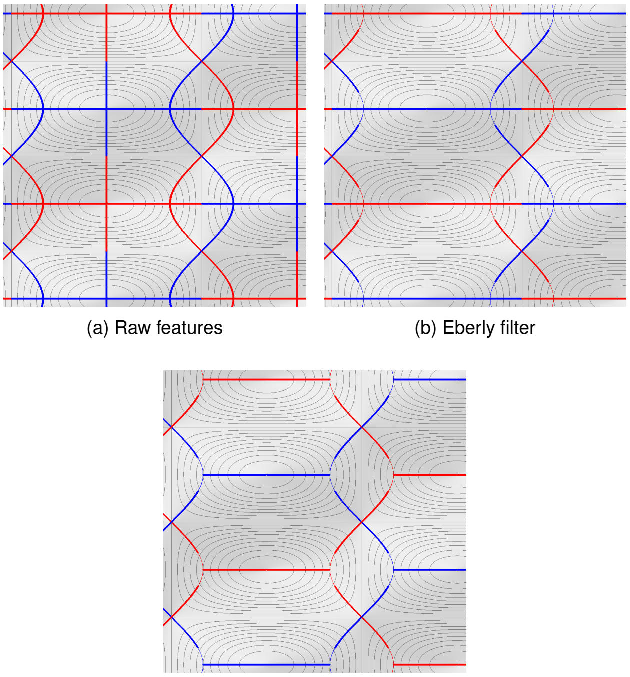
3 ONE-DIMENSIONAL HEIGHT RIDGES

For one-dimensional height ridges, (5) is exactly the *parallel vectors* condition [18]

$$\mathbf{Hg} = \lambda \mathbf{g} \quad \text{for some } \lambda \in \mathbb{R}. \tag{9}$$

This means that the set $\mathscr{R}_1$ of raw feature points of one-dimensional ridges can be found by collecting the points where the two derived vector fields $\mathbf{g}$ and $\mathbf{Hg}$ are parallel. Especially in higher-dimensional spaces ($n > 2$), this test for parallelism is more efficient than the explicit computation of eigenvalues and eigenvectors. The latter are still required for evaluating (2) and (3) or (4) but only on the lower-dimensional manifold $\mathscr{R}_1$.

The system (9) has n equations for $n+1$ unknowns, namely n spatial coordinates and λ. For a numerical solution, the intersection points with hyperplanes (e.g. cell boundaries) can be computed, this way getting rid of one coordinate. In $\mathbb{R}^2$ this is a simple contour extraction (see Section 5). In $\mathbb{R}^3$ it is tempting to replace (9) by $\mathbf{Hg} \times \mathbf{g} = \mathbf{0}$, which is however a linearly dependent set of equations. Algorithms for solving (9) on linear and bilinear cell faces are described in [18]. These can be extended to higher dimensions.



(a) Raw features

(b) Eberly filter

(c) Contour distance filter

Figure 1: Height function $f(x,y) = cos(ax)cos(by)$ with ridges (red) and valleys (blue). (a) Raw features. (b) Subset accepted by F_E, or equally, F_L. (c) Subset accepted by F_C. In (b) and (c), thin lines mean: rejected by F_{45°.

3.1 The Angle Criterion

Both height ridge definitions tacitly assume that the ridge is roughly aligned with the direction of the gradient $\mathbf{g}$, i.e. with the slope line. However, point-wise definitions cannot enforce such a condition, therefore it is no surprise that there exist examples of curves which are valid height ridges in the sense of the definition, but which have a large angle between their tangent and $\mathbf{g}$. It is therefore advisable to define a filter for removing such points from $\mathscr{R}_1$. This criterion for the feature alignment can be used as another filter F_α for post-processing the set $\mathscr{R}_1$. It involves the choice of a threshold angle, for which 45° is a good empirical value. In Figure 1(b,c) this filter removes parts of the raw feature that run almost along a contour line and that are accepted by all three other considered filters.

For a numerical computation of F_α the tangent of the feature line can be estimated from a few samples. This technique was used in [18]. In the simple case where $n = 2$, the feature tangent can be calculated analytically, as will be shown in Section 5. It is also possible to generalize the angle criterion to k-dimensional height ridges in n-space by using the angle between the tangent space of the ridge and the linear space spanned by the eigenvectors $\mathbf{y}_{n-k}, \cdots, \mathbf{y}_{n-1}$. The angle between linear subspaces is defined in [7].

3.2 The Contour Distance Criterion

The identity $\nabla \mathbf{g}^2 = 2\mathbf{Hg}$ reveals a different interpretation of (9), namely as a condition for an extremum of $\mathbf{g}^2$ along the height contour. That means that the distance of two (infinitesimally close) height contours assumes a maximum, a minimum, a flat spot, or (for $n > 2$) a saddle. From an intuitive point of view, however, this distance should be maximal, both for ridges and for valleys. This suggests a filter F_C which tests for a maximal distance of height contours.

4 Height Ridges of Co-Dimension One

If $k = n - 1$, there is a single equation in (1) to be satisfied. A straight-forward method to compute such ridges is to find zero crossings of $\mathbf{g} \cdot \mathbf{y}_1$ on grid edges, where $\mathbf{y}_1$ is the explicitly computed eigenvector belonging to the eigenvalue λ_1, with the ordering defined by either (3) or (4). A problem is, however, that the eigenvectors at the nodes of a cell can be inconsistently oriented. The *Marching Ridges* algorithm by Furst et al. [6] uses principal component analysis to achieve per-cell consistency of the eigenvectors, while Kindlmann et al. [13] track rotation of the eigenvectors along sub-sampled edges. But both of these schemes are problematic in cells which contain a degenerate point of the tensor field $\mathbf{H}$, i.e. a point where the two eigenvalues coincide. At a degenerate point, the ordering of eigenvalues changes, therefore the eigenvector associated with the i-th largest eigenvalue “jumps by 90 degrees”.

Alternatively, raw height ridges can be computed by using (5) which for $k = n - 1$ simplifies to

$$d = \det\left(\mathbf{H}^0\mathbf{g}\,|\cdots|\,\mathbf{H}^k\mathbf{g}\right) = 0. \tag{10}$$

In other words, raw height ridges of co-dimension 1 are simply the zero contours of a derived scalar field d and can therefore be computed with a n-dimensional version of the Marching Cubes algorithm [16].

5 Height Ridges in $\mathbb{R}^2$

In the special case $n = 2$, $k = 1$, the set $\mathscr{R}_1$ of raw feature points is by (10) the zero contour of the derived field

$$d = \det(\mathbf{g}\,|\mathbf{Hg}\,). \tag{11}$$

Because of this, a property of $\mathscr{R}_1$ is that it contains all critical points of $\mathbf{g}$ (where $\mathbf{g} = \mathbf{0}$) and for $n = 2$ also all degenerate points of $\mathbf{H}$ (where $\mathbf{H}$ is a pure scaling matrix).

Having the raw feature points, the next step is to classify them as either ridge points or valley points. For this, the eigenvalues of $\mathbf{H}$ associated with the directions of the gradient $\mathbf{g}$ and co-gradient $\mathbf{c} = (-g_2, g_1)$ are needed. If the gradient $\mathbf{g}$ is nonzero, these are simply $\lambda_g = \frac{\mathbf{Hg}}{\mathbf{g}}$ and $\lambda_c = \frac{\mathbf{Hc}}{\mathbf{c}}$, where the division of parallel vectors is defined component-wise, and is done by selecting the largest absolute component of the denominator. Now the sign of λ_c determines whether the point is on a ridge (negative), valley (positive) or neither (zero). Where $\mathbf{g} = \mathbf{0}$, it is sufficient to know the type of critical point, which is given by the determinant and trace of $\mathbf{H}$. If the point is a local maximum or minimum, it is classified as a ridge or valley point, respectively. If it is a saddle, it is a crossing of a ridge and a valley.

With the eigenvalues λ_g and λ_c, it is now possible to apply the tests (3) or (4) in order to eliminate feature points violating the respective height ridge definition. Critical points are always part of the ridge or valley, for both Eberly's and Lindeberg's definition. However, Lindeberg's definition does not allow for a crossing of a ridge and a valley, except in the case of equal absolute eigenvalues. Therefore, by Lindeberg's definition, a saddle point is either a ridge or a valley point.

The feature normal can be derived from (11) as $\mathbf{N} = \nabla d$ which is in explicit notation:

$$\begin{aligned}\mathbf{N}_1 &= f_{xxy}\left(f_x^2 - f_y^2\right) + \left(f_{xyy} - f_{xxx}\right) f_x f_y \\ &\quad + f_{xy}\left(f_{xx} + f_{yy}\right) f_x - \left(2f_{xy}^2 + f_{xx}^2 - f_{xx}f_{yy}\right) f_y \\ \mathbf{N}_2 &= f_{xyy}\left(f_x^2 - f_y^2\right) + \left(f_{yyy} - f_{xxy}\right) f_x f_y \\ &\quad - f_{xy}\left(f_{xx} + f_{yy}\right) f_y + \left(2f_{xy}^2 + f_{yy}^2 - f_{xx}f_{yy}\right) f_x.\end{aligned} \tag{12}$$

The feature tangent is then $\mathbf{T} = (-\mathbf{N}_2, \mathbf{N}_1)$. From this, an analytic expression for the filter F_α (with a threshold angle of α) can be derived. By comparing the projections of $\mathbf{T}$ onto $\mathbf{g}$ and $\mathbf{c}$, this is

$$\frac{|\mathbf{T}\cdot\mathbf{c}|}{|\mathbf{T}\cdot\mathbf{g}|} = \frac{|\mathbf{N}\cdot\mathbf{g}|}{|\mathbf{N}\cdot\mathbf{c}|} \leq \tan\alpha. \tag{13}$$

Finally, in the case $n = 2, k = 1$, the contour distance filter F_C can be expressed analytically. Because of $d = \det(\mathbf{g}\,|\mathbf{Hg}\,) = \mathbf{c}\cdot\mathbf{Hg} = \frac{1}{2}\mathbf{c}\cdot\nabla\left(\mathbf{g}^2\right)$, a negative (positive) value of d means that the contour distance increases (decreases) along the oriented contour line, where the orientation is given by the co-gradient $\mathbf{c} = (-g_2, g_1)$. Hence, at a raw feature point, we can conclude that the contour distance is maximal if and only if the gradient of d has a positive projection onto $\mathbf{c}$. That means, the contour distance filter can be written as:

$$\mathbf{N}\cdot\mathbf{c} > 0. \tag{14}$$

6 Implementation Details

Having introduced height ridges and filters for analytically given fields, we describe now their numeric implementations for discretized fields.

6.1 Raw Feature Extraction

Raw feature extraction can be done with standard methods for the practically most interesting cases of k-dimensional ridges in n-space. If the ridge to be extracted has co-dimension one, this amounts to solving an isocontour problem where many methods are available for 2-space and 3-space. For one-dimensional ridges in 3-space or higher, the parallel vectors method can be used as described in [18]. It computes the set of intersection points with cell faces and from these it constructs the feature lines. The case of general k and n requires an extension of that method. This can be done in principle by solving a system of algebraic equations. However, for large k and n, computing the full set of raw features is too much of an overhead, which in practice limits the general approach to small values of k and n.

Both algorithms, isocontour and parallel vectors, use implicitly or explicitly the two vector fields $\mathbf{g}$ and $\mathbf{Hg}$, precomputed on grid nodes. It is, however, more accurate to estimate the derivatives $\mathbf{g}$ and $\mathbf{H}$ directly at the given point instead of using interpolation. Therefore, quality can be improved by post-processing the intersection points using bisection or Newton iteration.

Once the intersection points are determined, they need to be connected to (discrete representations of) k-dimensional manifolds. Connecting is done on a cell-by-cell basis, and depending on the number of intersection points there may be several choices for connecting them within a given cell. In the Marching Cubes type algorithms these ambiguities are resolved in a simple systematic way, while the parallel vectors algorithm orders the intersection points by the ratio of the lengths of the two vectors and then connects pairs of points. In the setting of general k and n the connecting step would have to be defined, too.

6.2 Implementation of the Filters

The filters F_E and F_L, and can be evaluated point-wise if first and second derivatives of the height function can be estimated. In the special case of one-dimensional ridges in 2-space, also the filters F_α and F_C can be computed point-wise from derivatives, but this requires third derivatives. A way to avoid third derivatives in the angle filter is to estimate the feature tangent from three successive samples on the raw feature line. It is more difficult to avoid third derivatives in the contour distance filter. We sampled the gradient magnitude on nearby points on the isocontour (actually the isosurface in 3-space) and observed that for good results it is in fact important to place these samples not just on the normal plane of the gradient vector, but effectively on the isocontour.

Depending on the application, more filters can be useful. In the case of one-dimensional ridges, it often makes sense to discard feature lines that are below a certain minimal length. Obviously, the length should be measured after the point-wise filters F_E, F_L, F_α and F_C have been applied. But for a more robust algorithm, it is advisable to tolerate some small number of consecutive exceptions. The overall algorithm, if this type of length filter is used, is then as shown in (1). Typical values producing good results are $m_1 = m_2 = 3$ and $m_3 = 20$.

Algorithm 1 Height ridge extraction and filtering

```
for all grid cells $c$ do
    compute intersection points of raw feature lines with $c$
    connect these vertices to raw feature line segments
    evaluate filter criteria on all vertices
    label vertices as valid if they pass all filters
end for
connect line segments to polylines  // possibly closed
for all polylines $l$ do
    search for a sequence of $m_1$ valid vertices
    extend the sequence forward and backward as long as vertices are valid or the number of consecutive exceptions is less than or equal to $m_2$
    after ending this extension step, remove any exception vertices at the end
    accept the obtained polyline if it contains at least $m_3$ vertices
end for
```

6.3 Numerical Derivatives

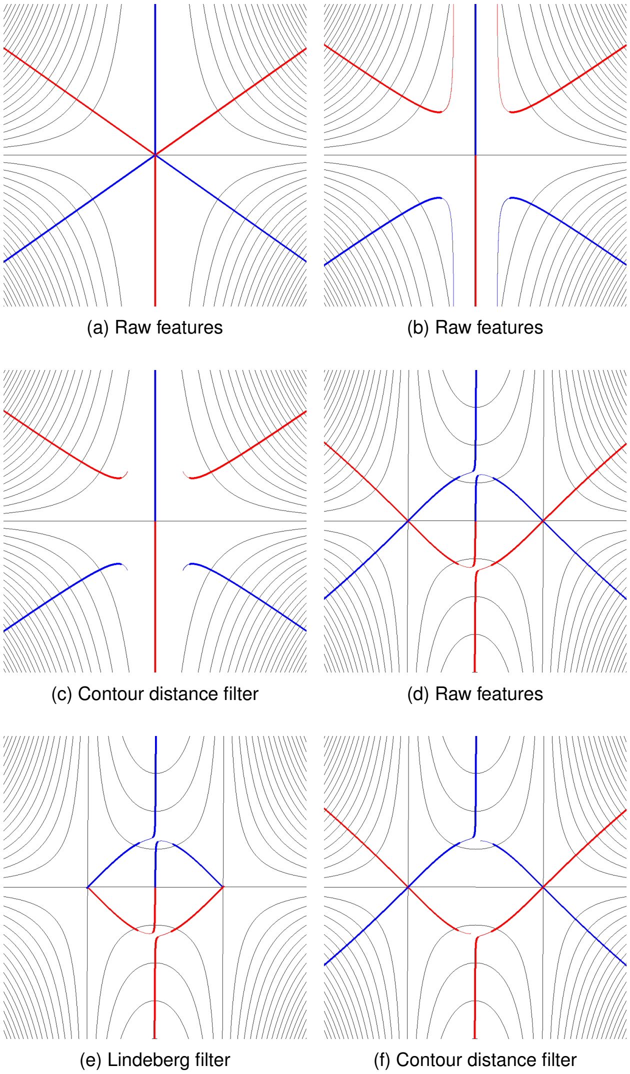
The methods described in this paper require up to third order derivatives of the given height function. In most cases, analytic derivatives are not available, so numerical derivatives must be computed. While the quality of derivatives is important, the method can be freely chosen, depending on the discretization type of the data. In the 2D example (Section 7.2) we computed derivatives by convolving the data with derivatives of the Gaussian. Since in this example, we computed the angle and contour distance filters based on third derivatives, such high-quality numerical derivatives were needed. In the 3D example (Section 7.3) we did gradient estimation by least-squares fitting of a linear function to the data. We used this simple method, which is easily applicable to 3D unstructured grids, because we computed only second derivatives (and computed the feature tangent using three consecutive vertices).

7 Evaluation of Height Ridge Filters

The current use of height ridges in scientific visualization focuses on 3D data. But to allow for a better visual assessment of the different filters, we first use a synthetic and a numeric 2D height field before ending with a numeric 3D scalar field.

7.1 Synthetic Data: "Monkey Saddle" Example

The "Monkey saddle" example $f(x,y) = x^2y$ taken from Eberly's book [5] has as raw features three ridges and three valleys joining in a point, see Figure 2a. These are completely accepted by the filters F_E, F_C and F_{45°, but completely rejected by the more restrictive F_L. In a perturbed version $f(x,y) = x^2y + 0.02y$ a single ridge-valley transition remains and other four (raw) feature curves do not intersect. F_E accepts the full raw feature set, including clear false positives (b), F_C rejects them (c), while F_L accepts only the vertical ridge. In the differently perturbed field $f(x,y) = x^2y - 0.2y$, the monkey saddle breaks up in two ordinary saddles and a ridge-valley transition. The angle filter F_{45° rejects parts around the near T-junction. F_E accepts all (d), F_L just a subset (e), and F_C accepts all but the virtual solution in the center where contours have minimal instead of maximal distance.


(a) Raw features

(b) Raw features

(c) Contour distance filter

(d) Raw features

(e) Lindeberg filter

(f) Contour distance filter

Figure 2: “Monkey saddle” example $f(x,y) = x^2y$ from [5] with ridges (red) and valleys (blue). Thin lines represent segments rejected by the angle filter F_{45°. (a) Unperturbed field. (b,c) Perturbed field $f(x,y) = x^2y + 0.02y$. (d-f) Perturbed field $f(x,y) = x^2y - 0.2y$.

7.2 Application: Ridges in Topographic Data

As numerical test data we use a topographic dataset which is available as a free sample from `http://www.swisstopo.ch/en/download/testdata/height/dhm25` as the file `mma125.xyz` containing a 161x161 grid at 25 m resolution (Albis region, south of Zurich, approx. here: `http://maps.google.com/maps?ll=47.27,8.507&z=14`). By (11), raw features are obtained as the zero contour of d. In order to minimize the effect of numerics in our comparison, we used high-quality derivatives, obtained by convolution with derivatives of a Gaussian with $\sigma = 25$ m and a cutoff radius of 5σ, and we iteratively refined contour samples using bisection of grid edges. In practical applications, it is usually necessary to add a filter for removing “weak” features caused by noise. We did not do this here with the exception of the masking of flat regions (the lake). Figure 3 shows the ridges and valleys with the contour distance and F_{45° filters applied. Perhaps noteworthy are the many (top-down) valley-ridge transitions. As already noticed by Jordan [12] these indicate alluvial fans (debris deposits). In the close-ups in Figure 4, thin lines represent curve segments removed by F_{45°. This filter produces acceptable results, even when applied as the only one to the raw features (a). False negatives can practically be excluded, but there are clear false positives near the main ridge. While F_L is obviously too restrictive (b), F_E produces good results (c) if combined with F_{45° (thick lines only), except for the many false positives, running mostly about parallel to actual features. F_C corrects this (d), and yields otherwise similar results.

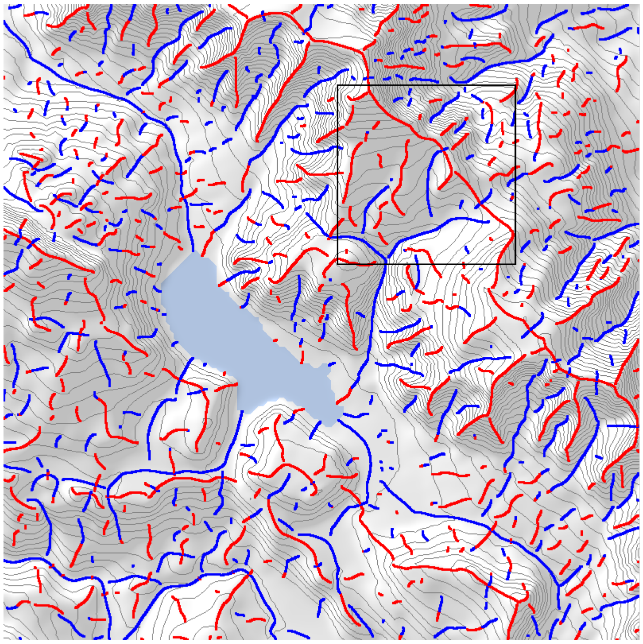
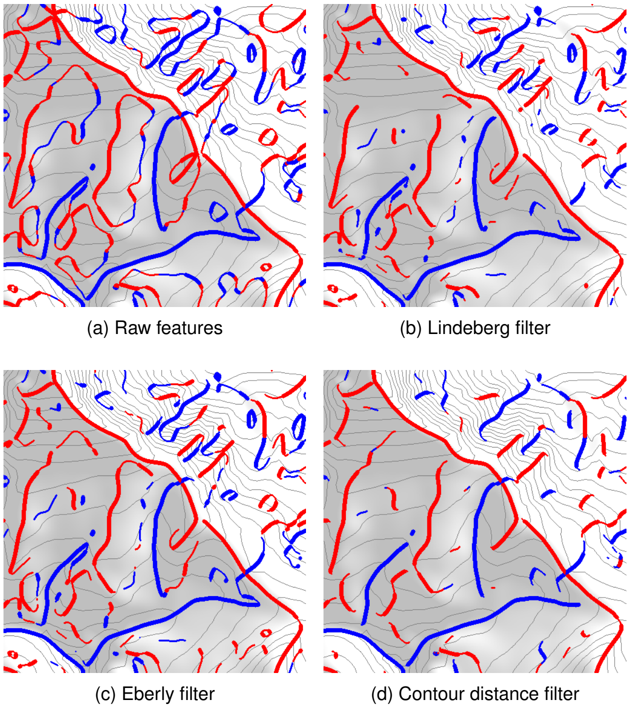


Figure 3: Ridges (red) and valleys (blue), with contour distance and 45° filters applied.


(a) Raw features

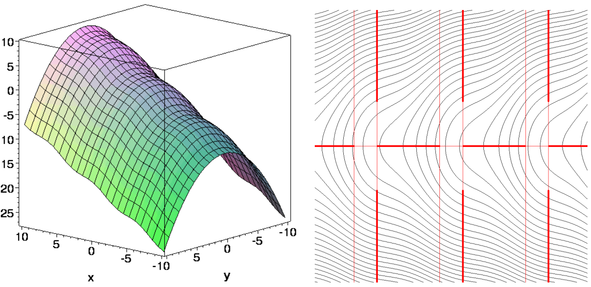
(b) Lindeberg filter

(c) Eberly filter

(d) Contour distance filter

Figure 4: (a) Close-up of Figure 3. Comparison of filters, and combination with F_{45° (thin lines).

The remaining problem that can be identified in Figure 4d are false negatives, i.e. gaps in feature curves. One reason for gaps introduced by F_C (and also F_E) are smaller-scale transversal ridges as is illustrated in Figure 5. This type of gaps could be closed by applying tolerant filters as was explained in Section 6.2. However, this pattern can hardly be spotted in this application. Most gaps are either introduced by the angle filter F_α or they are due to disjoint raw features. In the case of disjoint raw features, gaps cannot be closed easily. Many gaps of this type appear where ridges should form a T-junction, which is not possible for height ridges since they cannot branch as Damon [4] showed. Therefore, we decided not to apply any length filtering at all, with the effect that even very small ridges and valleys can be seen in the final visualization.



(a) (b)

Figure 5: (a) Height function $f(x,y) = x - \frac{1}{6}y^2 + \frac{1}{2}\cos(x)$. Along the ridge on the x-axis, eigenvectors are axis-aligned. Eigenvalues are $-\frac{1}{2}\cos(x)$ and $-\frac{1}{3}$ for the x and y directions, respectively. (b) Raw features and effect of F_C (thick lines). The filter F_C, as well as F_E, would interrupt the main ridge around the points $((2k+1)\pi, 0)$ where $-\frac{1}{2}\cos(x)$ falls below $-\frac{1}{3}$. F_L would create additional gaps where it exceeds $\frac{1}{3}$, that is, around the points $(2k\pi, 0)$.

7.3 Application: 1D Ridges of Negative Pressure in 3D CFD Data

For a 3D application we chose the CFD simulation of a Pelton water turbine. The region of interest is near the first one of six bifurcations of the distributor ring, which we already visualized in [19]. We selected the pressure data channel and computed one-dimensional ridges and valleys, as is shown in Figure 6.

As can be expected in 3D, the set of raw features (a) is significantly reduced by applying either the Lindeberg (b) or Eberly filter (c). The latter is further reduced by applying the 45° filter (d) and by requiring a minimum length of 15 vertices (e). The remaining set, which is almost identically obtained also by using F_L instead of F_E, consists of four feature lines, one of which coinciding with a vortex core line. The second vortex core line is not captured by any (raw or filtered) pressure valley line. Vice versa, three of the pressure valley lines are missed by the Sujudi-Haimes vortex core line method, meaning that they are at least no longitudinal vortices.

8 Conclusion

Being able to compute the raw height ridge points without using eigenvalues makes the overall computation efficient, at least in low-dimensional spaces. This approach also has the flexibility that one or more filters can be chosen to select the final feature points. The synthetic height fields and the terrain data used in this study suggest that the newly introduced contour distance filter is the best choice. The numerically less demanding filter based on Eberly's definition produces acceptable results, but only if combined with the angle filter, as we also observed in our example of 1D ridges in 3-space. Further investigation is, however, needed in the case of 2D ridges. Finally, we believe that further research is also needed toward a more satisfactory representation of branching.

References

[1] A. J. C. Barré de Saint-Venant. Surfaces plus grande pente constitues sur des lignes courbes. *Bulletin de la soc. philomath. de Paris*, pages 24–32, March 1852.

[2] A. G. Belyaev, E. V. Anoshkina, and T. L. Kunii. Ridges, ravines, and singularities. In A. T. Fomenko and T. L. Kunii, editors, *Topological Modeling for Visualization*, chapter 18, pages 375–383. Springer, 1997.

[3] P. E. Breton (de Champ). Note sur les lignes de faîte et de thalweg. *Comptes rendus hebdomadaires des séances de l'académie des Sciences*, 39:647–648, 1854.

[4] J. Damon. Properties of Ridges and Cores for Two-Dimensional Images. *J. Math. Imaging Vis.*, 10(2):163–174, 1999.

[5] D. Eberly. *Ridges in Image and Data Analysis*. Computational Imaging and Vision. Kluwer Academic Publishers, 1996.

[6] J. D. Furst and S. M. Pizer. Marching Ridges. In *Proceedings of the IASTED International Conference, Honolulu*, pages 22–26, 2001.

[7] H. Gunawan, O. Neswan, and W. Setya-Budhi. A Formula for Angles between Subspaces of Inner Product Spaces. *Contributions to Algebra and Geometry*, 46(2):311–320, 2005.

[8] G. Haller. Lagrangian coherent structures from approximate velocity data. *Phys. Fluids*, 14:1851–1861, 2002.

[9] R. M. Haralick. Ridges and Valleys on Digital Images. *Computer Vision, Graphics, and Image Processing*, pages 28–38, 1983.

[10] J. Helman and L. Hesselink. Representation and Display of Vector Field Topology in Fluid Flow Data Sets. *Computer*, 22(8):27–36, 1989.

[11] V. Interrante, H. Fuchs, and S. M. Pizer. Enhancing Transparent Skin Surfaces with Ridge and Valley Lines. In *IEEE Visualization*, pages 52–59, 1995.

[12] C. Jordan. Sur les lignes de faîte et de thalweg. *Comptes rendus hebdomadaires des séances de l'académie des Sciences*, 74:1457–1459, 1872.

[13] G. Kindlmann, X. Tricoche, and C.-F. Westin. Anisotropy Creases Delineate White Matter Structure in Diffusion Tensor MRI. In *Ninth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'06)*, Lecture Notes in Computer Science 4190, pages 126–133, October 2006.

[14] J. J. Koenderink and A. J. van Doorn. Two-Plus-One-Dimensional Differential Geometry. *Pattern Recognition Letters*, 15(5):439–443, May 1994.

[15] T. Lindeberg. Edge Detection and Ridge Detection with Automatic Scale Selection. *International Journal of Computer Vision*, 30(2):117–156, 1998.

[16] W. E. Lorensen and H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics*, 21(4):163–169, July 1987.

[17] P. Majer. *A Statistical Approach to Feature Detection and Scale Selection in Images*. PhD thesis, Georg-August Universitt Göttingen, 2000.

[18] R. Peikert and M. Roth. The "Parallel Vectors" Operator - A Vector Field Visualization Primitive. In *IEEE Visualization*, pages 263–270. IEEE Computer Society, 1999.

[19] R. Peikert and F. Sadlo. Topology-guided Visualization of Constrained Vector Fields. In H. Hauser, H. Hagen, and H. Theisel, editors, *Topology-based Methods in Visualization*, pages 21–34. Springer-Verlag, 2007.

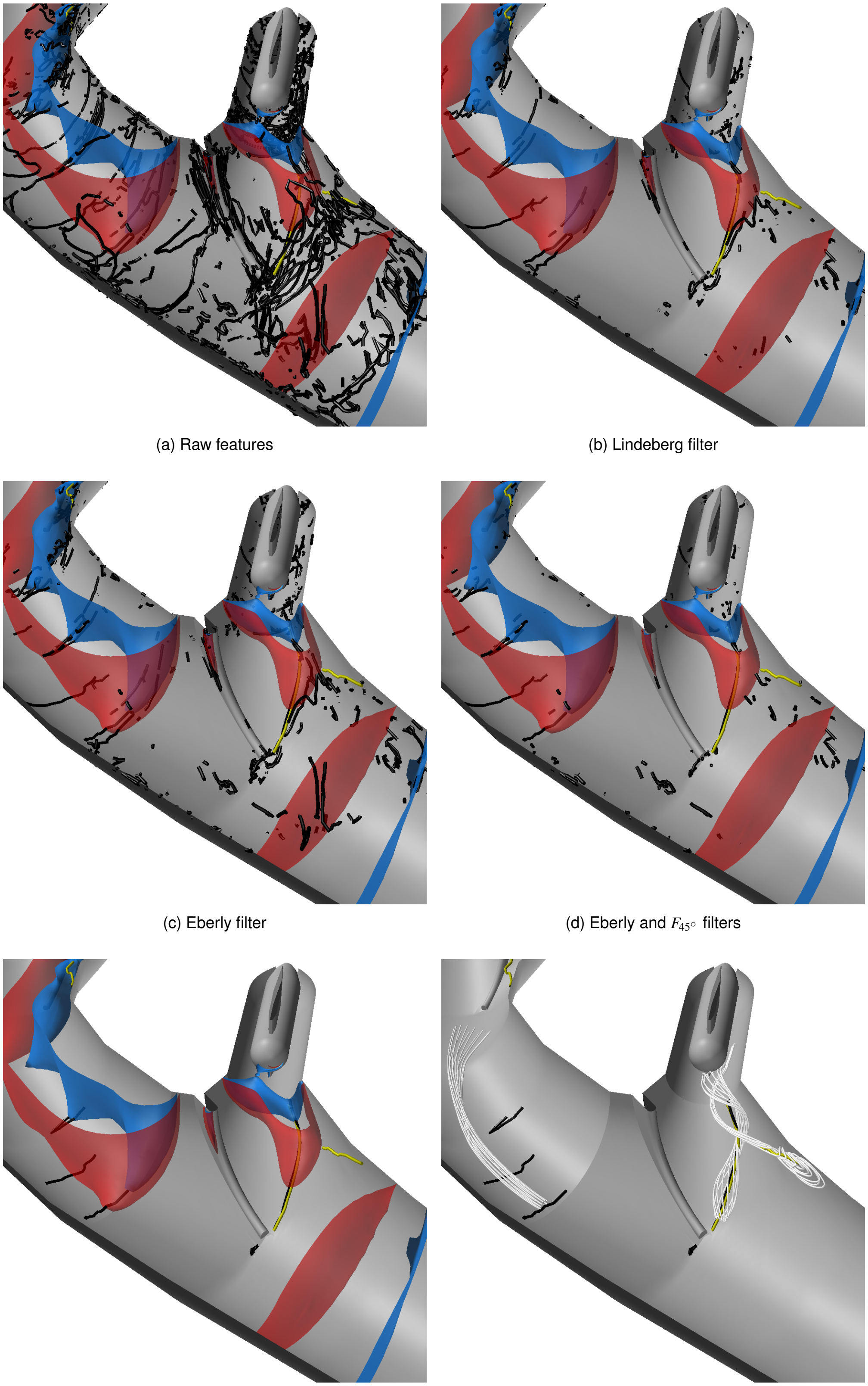
[20] I. R. Porteous. *Geometric Differentiation*. Cambridge Univ. Press, Cambridge, UK, 1994.

[21] F. Sadlo and R. Peikert. Efficient Visualization of Lagrangian Coherent Structures by Filtered AMR Ridge Extraction. *IEEE Trans. Vis. Comput. Graph.*, 13(6):1456–1463, 2007.

[22] J. Sahner, T. Weinkauf, N. Teuber, and H.-C. Hege. Vortex and Strain Skeletons in Eulerian and Lagrangian Frames. *IEEE Trans. Vis. Comput. Graph.*, 13(5):980–990, 2007.

[23] J. Serrat, A. Lopez, and D. Lloret. On Ridges and Valleys. *IPCR*, 04:4059, 2000.

[24] S. C. Shadden, F. Lekien, and J. E. Marsden. Definition and properties of Lagrangian coherent structures from finite-time Lyapunov ex-



(a) Raw features

(b) Lindeberg filter

(c) Eberly filter

(d) Eberly and F_{45° filters

(e) Eberly, F_{45°, length filters

(f) same, with streamlines

Figure 6: Steady simulation of flow in a Pelton distributor ring. Black tubes are valley lines of pressure, yellow tubes are vortex core lines (for comparison). Isosurfaces of pressure are shown in red (high) and blue (low). Two out of three streamline bundles confirm longitudinal vortices.

ponents in two-dimensional aperiodic flows. *Physica D Nonlinear Phenomena*, 212:271–304, Dec. 2005.

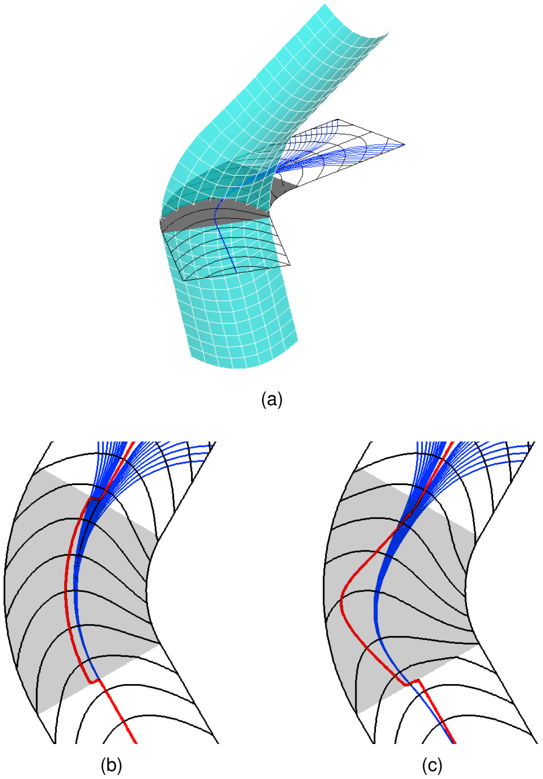
[25] C. Steger. Subpixel-Precise Extraction of Watersheds. In *ICCV (2)*, pages 884–890, 1999.

[26] R. Strawn, D. Kenwright, and J. Ahmad. Computer Visualization of Vortex Wake Systems. In *American Helicopter Society 54th Annual Forum*, May 1998.

[27] D. Sujudi and R. Haimes. Identification of Swirling Flow in 3D Vector Fields. Technical Report 95-1715, AIAA, 1995.

Appendix A. The "Curved Gutter" Example

The"curved gutter" example [14] consists of a parabolic profile that is in a helical motion rotated and at the same time elevated. We added two straight segments with matching profile at both ends. The watercourse (in Figure 7 indicated by the converging slope lines (blue)) follows the deepest points in radial sections of the curved part. The height ridge (red) misses them and delivers the "correct" result only in the straight parts. Therefore, the watercourse can be argued to be better than the height ridge because it respects the helical symmetry of the height field. However, the example can be changed slightly to become an example that favors the height ridge. For this purpose the curved section of the gutter is replaced by a blend of the two height fields representing the two straight gutter segments, with blending factors depending linearly on the angle. In this height field, the height ridge follows the deepest points of the normal sections in the straight parts, while the watercourse misses them in the lower straight part.


(a)

(b)

(c)

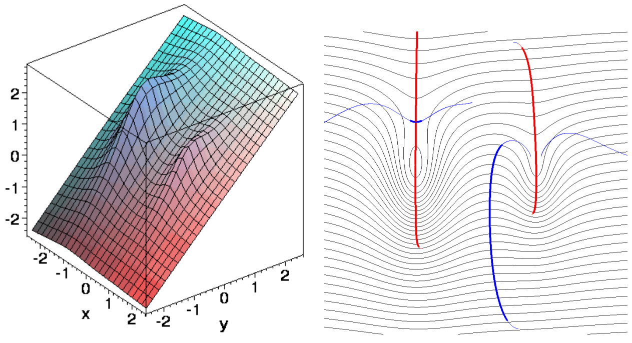
Figure 7: (a) "Curved gutter" example with two straight segments added. (b) Slope lines (blue) converging to the watercourse. Height ridge (red) has a radial offset. (c) "Blended gutter" obtained by replacing the curved part (shaded region) by a linear blend of the two height fields of the straight parts. Watercourse (blue) converges to, but is not identical with, the obvious valley line in the lower segment.

Appendix B. Ridges without Critical Points

Height ridges and watersheds are fundamentally different concepts both having their values for visualization. While height ridges cannot describe the drainage pattern of a terrain, watersheds cannot capture some types of visually evident ridges, because by definition, a watershed requires a saddle from which integration can start. As an example, let us consider the height field

$$f(x,y) = y + \frac{2}{1+4(x+1)^2+y^2} + \frac{1}{1+8(x+1)^2+2y^2}$$

which is a ramp with two ridges superimposed. Only the larger ridge contains critical points (a saddle and a maximum). The smaller ridge is therefore not a watershed, even though it has the visual appearance of a ridge (see Figure 8). It can be made a watershed by adding a pair of a maximum and a saddle along the ridge, for which an arbitrary small perturbation is sufficient. Both ridges are valid height ridges in the sense of F_E and also F_C.



(a)

(b)

Figure 8: Pair of a watershed and non-watershed ridge. (a) Graph of the height field, (b) Contours, and extracted ridges and valleys, filtered with F_C. Thin lines: rejected by $F_{45°}$.