

Enridged Contour Maps

Jarke J. van Wijk*

Alexandru Telea*

Eindhoven University of Technology

Abstract

The visualization of scalar functions of two variables is a classic and ubiquitous application. We present a new method to visualize such data. The method is based on a non-linear mapping of the function to a height field, followed by visualization as a shaded mountain landscape. The method is easy to implement and efficient, and leads to intriguing and insightful images: The visualization is enriched by adding ridges. Three types of applications are discussed: visualization of iso-levels, clusters (multivariate data visualization), and dense contours (flow visualization).

Keywords: Contours, mapping, height fields, multivariate visualization, flow visualization

1 INTRODUCTION

The display of scalar data defined over surfaces is possibly the most ubiquitous application of visualization. Typical examples are maps of temperature, height, and air pressure. Also, 3D data volume data are often visualized by taking slices, thereby reducing the dimension of the data to be visualized.

Many techniques are available for the visualization of scalar functions $f(x, y)$, each with its own strengths [11]. A natural way to interpret such data is as a *height field*, and hence to display them as a mountain landscape. A perspective view can be used, but this leads to occlusion of data. A top view on a shaded surface is more effective (fig. 1a). Structures can easily be discerned, as the human visual system is trained in the interpretation of shaded surfaces. Also, for the description of features typically 3D metaphors are used: ridges, peaks, valleys, and saddle-points. A drawback of such displays is that quantitative information is hard to extract: It is not easy to judge for instance which of the peaks is the highest. One solution is to map the scalar data to a grey shade or color (fig. 1b). Each color corresponds to a fixed level, the associated value can be read from a color map. Though popular, this method is not without problems and will not always lead to an improvement [4]. The popular rainbow color map can lead to false interpretations, grey shade images look fuzzy and not attractive. Another solution to visualize quantitative information is to use contours or iso-lines (figure 1c). They enable the viewer to establish precise quantitative information locally, but do not show the structure as clearly as shaded surfaces. Furthermore, labels are required to retrieve the magnitude of the function at some arbitrary point at the contour.

Each method has its own strengths and weaknesses, hence their combined use gives often the best results (fig. 1d). The price to be paid is that this requires several graphical means simultaneously: lines, shading, and color, which cannot be used anymore for the visualization of other information in the same image. A method that shows both qualitative and quantitative information using only one graphical channel would be helpful in such circumstances.

In this paper we present such a method. We use shaded mountain landscapes as the basic visualization technique, and show how

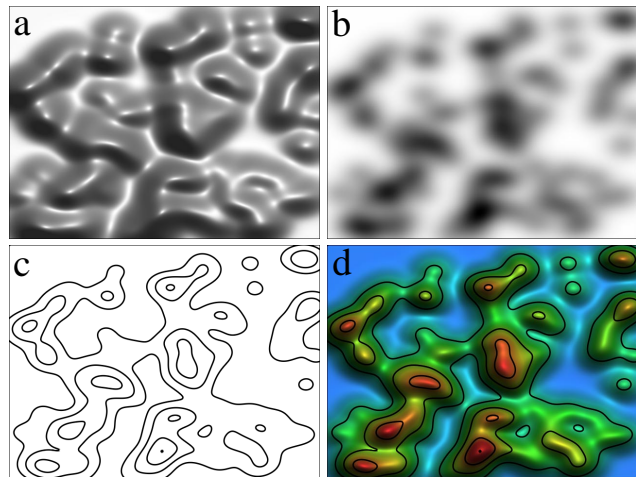


Figure 1: Visualization of scalar field

via non-linear transformations of the height quantitative information can be visualized more directly. We have used a similar concept before. In [10] we have shown how treemaps can be enhanced with hierarchical shading, in [6] we have used cushions to enhance various types of Voronoi diagrams. Here we show how a combination of shading and non-linear transformations aid in understanding scalar functions (section 2), 2D representations of clusters (section 3), and dense contour plots (section 4). Finally, we present some applications (section 5), and discuss extensions and summarize the results in section 6.

2 SCALAR DATA

Suppose we want to visualize a scalar function $f(\mathbf{x})$, with $\mathbf{x} \in \mathbb{R}^2$. Extension to the more general case, where the domain is a surface embedded in three-dimensional space, is straightforward, for simplicity we limit ourselves to the simplest case. A standard mapping of this function to a height $z(\mathbf{x})$ is via straightforward linear scaling

$$z(\mathbf{x}) = sf(\mathbf{x})$$

where s is a suitably chosen scale factor. We aim at emphasizing various levels of the height, and to this end we use a non-linear mapping

$$z(\mathbf{x}) = sf(\mathbf{x}) + sh g\left(\frac{f(\mathbf{x}) \bmod h}{h}\right)$$

with

$$g(x) = ax(1-x).$$

In other words, for succeeding intervals of f with width h we add a parabola. The factor a determines the height of the parabolas. Negative values of a give concave ridges, positive values give convex ridges. For values of $a \in [-1, 1]$ the mapping is monotonous. In

*Eindhoven University of Technology, Dept. of Mathematics and Computer Science, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. E-mail: {vanwijk, alext}@win.tue.nl

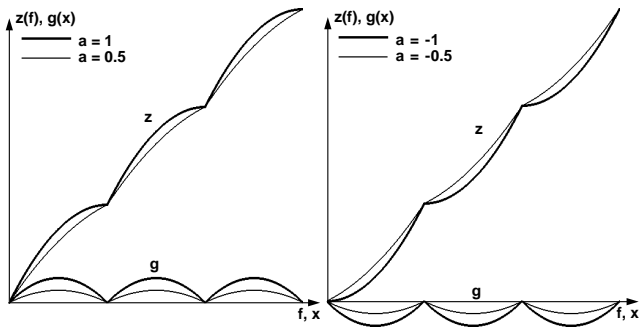


Figure 2: Mapping for various values of a

figure 2 the graphs of $g(x)$ and the graphs of the mapping of f to z are shown. The deviation from a linear mapping is small, especially for small values of a .

In figure 3 however, results for various values of a for a shaded image are shown. The following observations can be made:

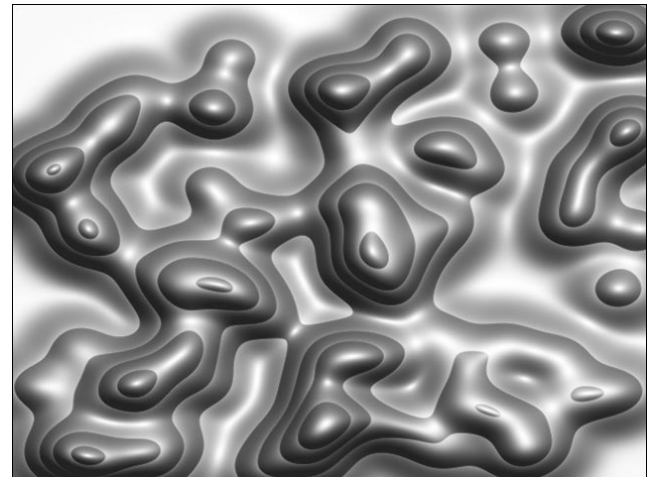
- The parameter a gives a good control over the strength of the banding effect. The user can easily tune this parameter to get just as much emphasis on the ridges as needed.
- Convex ridges are more pleasant and natural than concave ridges, the visual system is more used to convex than to concave shapes.
- The ridges show up very clearly, although the geometric deformation is fairly small. To illustrate the latter, figure 4 shows the value of $z(x)$ directly mapped to grey shades, and here the bands are hardly discernable. This effect can easily be explained. For the geometric version the intensity depends on the normal on the surface, i.e. the variation in the value of $f(x, y)$, and not on the value of $f(x, y)$ itself. Hence, the first order discontinuities of the height function show up much stronger than for direct mapping to grey shades.

A subtle effect is that the different bands seem to have a different grey shade value: for positive a , the closer to the tops, the darker they seem, while for negative a the reverse holds. This is an optical illusion, known as the Cornsweet effect [11]. The brain constructs the overall brightness of the bands based largely on edge contrast information. At a transition between two ridges the lower ridge is more horizontal for a positive a and hence brighter than the upper ridge, so the brain decides that the lower ridge must be brighter. Also, the visual system seems to use transitivity: If ridge 1 is brighter than ridge 2, and ridge 2 is brighter than ridge 3, then ridge 1 must be much more bright than ridge 3, etc.. We can verify that it is an illusion by looking at figure 3 from a distance. If the sharp transitions can not be observed, the ridges all appear to have the same color.

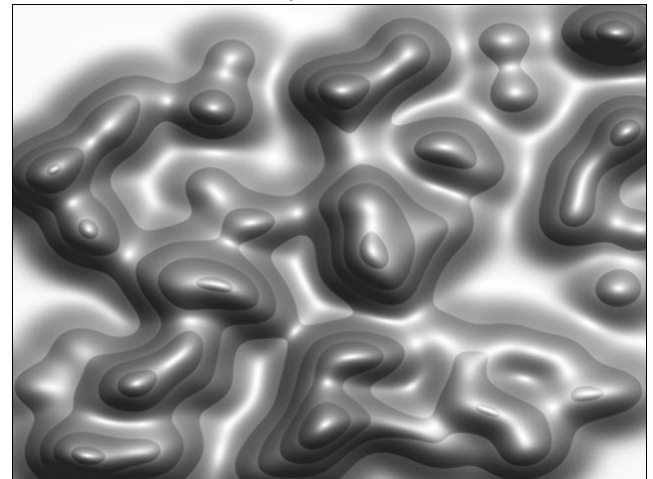
For the judgement of the absolute values this effect is very beneficial. Areas for which $f(x, y)$ is within a certain interval can easily be discerned, for the comparison of heights one often does not have to count the number of contour lines. Also, the dynamic range of the output device is exploited very well. Each band can in principle use the full range, while the described visual effect produces yet another series of grey shade bands.

A straightforward generalization is to use hierarchical ridges. Instead of a single parabola, multiple parabolas can be added with differing widths and heights. In figure 5 we used $a = 0.5$ for the major and $a = 0.25$ for the minor ridges.

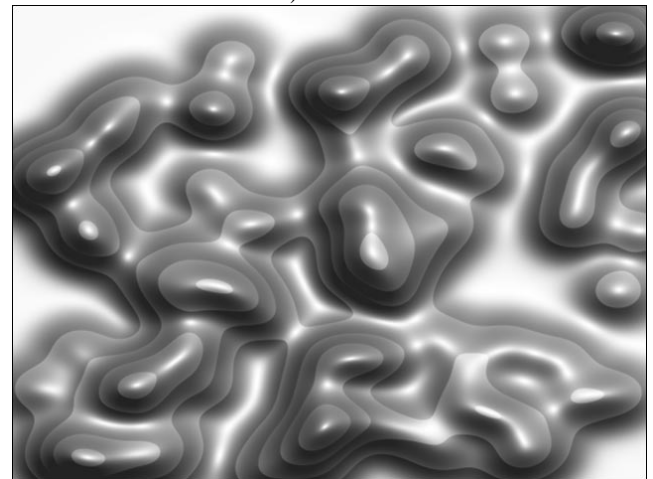
We have experimented with different profiles for the ridges. We have even implemented a version where the user can define the function $g(x)$ interactively by editing a spline. However, we found that



a) $a = 0.6$



b) $a = 0.3$



c) $a = -0.3$

Figure 3: Non linear mapped scalar field

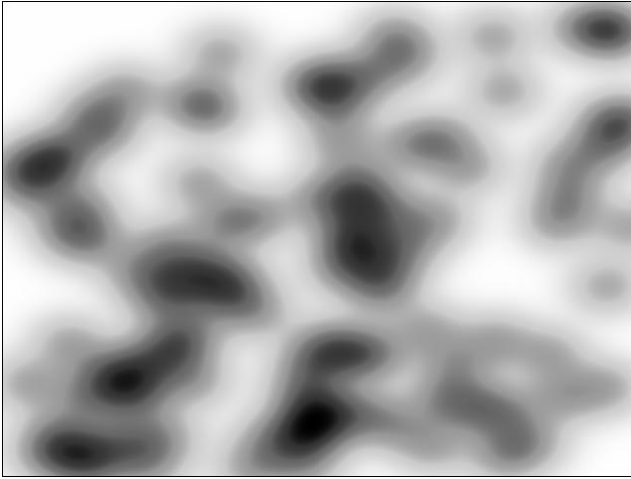


Figure 4: Non-linear mapping to grey values, $a = 0.6$

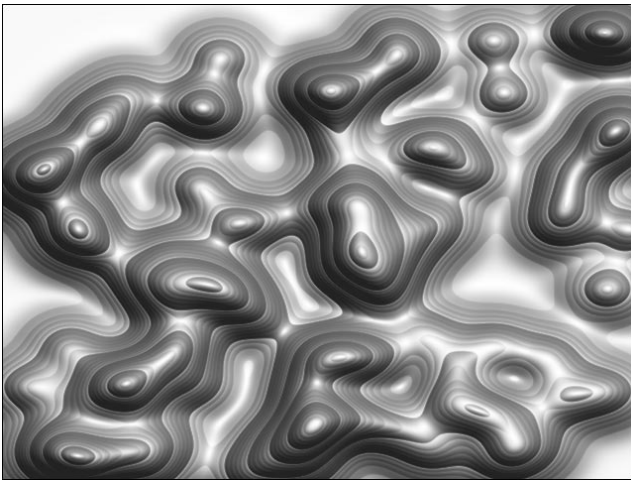


Figure 5: Hierarchical ridges

always the best results were achieved with a simple convex parabola, with one parameter a to control the strength of the effect.

3 CLUSTERS

Consider a large number of items (such as cities, persons, brands), where each item has a number of attributes (such as size, length, market-share). How can we visually get insight in such multivariate data? The use of scatterplots [2] is a popular method. Items are shown as points or icons, and projected on a 2D area, where the axes correspond to two different attributes. Scatterplot matrices enable the user to view more than two attributes. Another approach is to use Multi-Dimensional Scaling (MDS) [1]. The principle of MDS is to project a set of items with a number of attributes to a space of lower dimensionality (typically a plane), such that the mutual distances are preserved as well as possible. Many different techniques can be used to calculate optimal positions as well as many different measures for the distances between points in high-dimensional space.

In such plots two types of features are of particular interest. The first one is correlation. If the points fit more or less on a line or curve, this can indicate a linear or more complex relation between two variables. Second, we consider clusters. If the items fall apart into dis-

joint groups, the points in a scatterplot will cluster also into separate groups. Clusters often have a hierarchical character. Dependent on the threshold for the similarity measure used, small clusters merge into larger clusters.

A great example in this respect are the ThemeScapes of Wise et al. [12]. MDS is used here to visualize news archives. From the positions of the points, each representing an article, a height field is derived, which is visualized via a cartographically inspired, very well designed combination of shading, color, and contours. As a result, large numbers of closely related articles give rise to mountains. Another large class of examples are software visualization tools that map software systems to graphs, where the distance between components is related to their interdependence.

Hence, the visualization of clusters in multivariate data can be transformed into the same problem as discussed in the preceding section: Visualization of a function $f(\mathbf{x})$, where f represents a density. For this application however, it would be nice if we can aid the user in interpreting the multivariate data as a set of nested clusters.

An obvious way to realize this is via the use of contours, which by definition form hierarchically structured Venn-diagrams. A drawback here is that for complex contour sets the interior and exterior of contours can not be distinguished. An alternative is shown in figure 6. The test function used was defined by convolving a number of randomly positioned points, shown in red, with a Gaussian. For the visualization we used a variation on the method presented in the previous section. Instead of a parabola, we used a cubic mapping

$$g(x) = x(x - 1)(x - 2).$$

As a result, at the lower side of a band a steep wall is visible, while at the upper side the surface is horizontal and has zero curvature. Our aim was to visualize clusters as stacked drops of fluid. We show that this can be realized without a complex physical model, but instead just by a simple non-linear mapping. The hierarchy in the levels can easily be observed, without using color. Another strong visual effect is that minima and maxima can easily be distinguished.

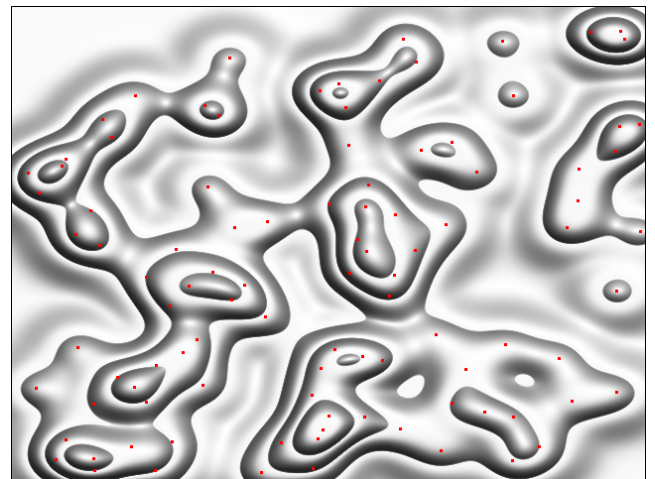


Figure 6: Nested clusters

A delicate aspect of the use of contours in general and also of our method, is that they suggest more than is contained in the data. A small change in the settings can lead to drastically different shapes. Hence, such methods should be embedded in an interactive system, such that the user can experiment with different values for the contour spacing.

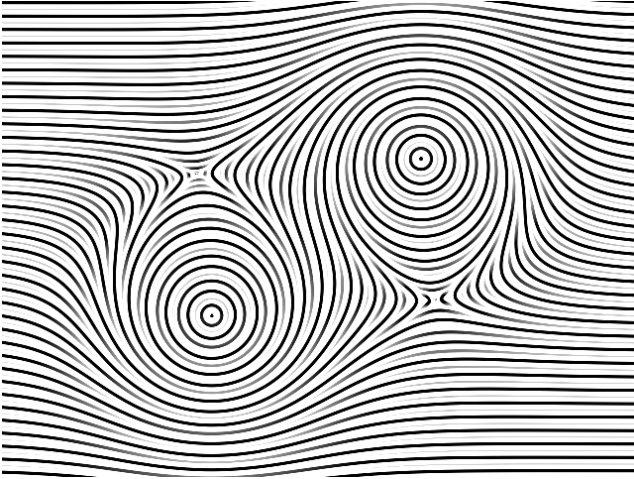


Figure 7: Dense contours

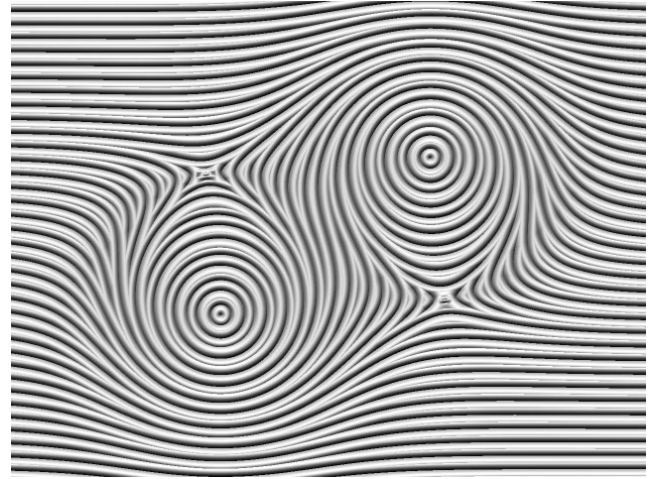


Figure 9: Shaded dense contours

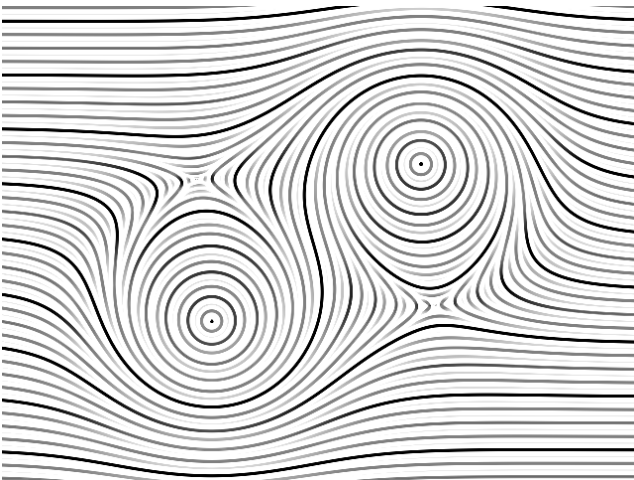


Figure 8: Hierarchical dense contours

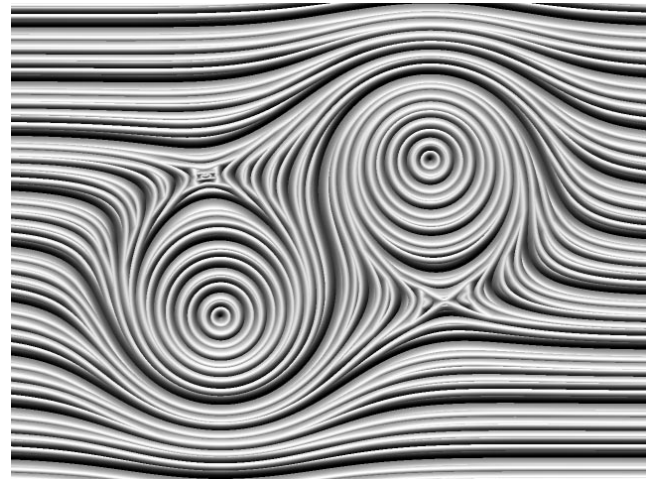


Figure 10: Hierarchical shaded dense contours

4 DENSE CONTOURS

Each point on a contour provides precise information on the magnitude of the value of the visualized function, as well as on the local change in the value, i.e. the gradient. The magnitude of gradients is shown via the density of the contours, as anybody who has used a map with height contours will know. A disadvantage here is that areas with a low magnitude of the gradient, i.e. more or less flat areas, large gaps lie between contours, and hence no information on local structure is shown.

Saito and Takahashi have introduced dense contours [5]. Contours are generated such that the distance between contours is constant in the space of the domain of the function. A related topic is the generation of streamlines for 2D flow fields. How can we generate streamlines such that the area is uniformly covered? Turk and Banks [7] use a complex optimization based method to realize this, a simpler method is given by Jobard and Lefer [3]. The generation of dense streamlines in general is inherently more complex than the generation of dense contours. However, when streamlines can be defined as contours, a simpler method can be used. For two-dimensional flows that are dominated by viscosity and where inertia can be ignored, the vector field can be modelled as $\mathbf{v} = [\partial\Psi/\partial y, -\partial\Psi/\partial x]$, where $\Psi(\mathbf{x})$ is the stream function. Con-

tourlines of $\Psi(\mathbf{x})$ are streamlines. However, the method we propose here is not limited to just stream function based flow fields. The only requirement here is that the contours of some function are streamlines of a given flow field. In [9] we have described how such functions can be calculated by prescribing a varying concentration at the boundaries, followed by a convection step.

We use an approach similar to Saito and Takahashi, and use shading and non-linear mapping again to improve the images. The calculation of dense contours proceeds as follows. We define a set of contour-lines with spacing h in the range of f as

$$L(h) = \{\mathbf{x} \in \mathbb{R}^2 \mid f(\mathbf{x}) = kh, k \in \mathbb{N}\},$$

and a hierarchy of sets as

$$L(b^i), i \in \mathbb{N},$$

where $b \in \mathbb{N}$ is a multiplication factor for the spacing from level i to level $i + 1$. Dependent on the gradient $\nabla f = [\partial f/\partial x, \partial f/\partial y]$ we select an appropriate level. Consider a line $\mathbf{q}(s)$, perpendicular on a contour:

$$\mathbf{q}(s) = \mathbf{x}_0 + s \nabla f / \|\nabla f\|, s \in \mathbb{R}$$

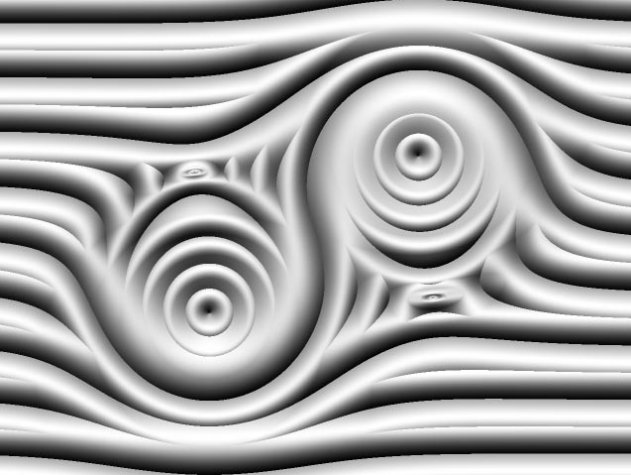


Figure 11: Wide ridges

Along this line it will hold by definition that

$$\|\Delta f / \Delta s\| = \|\nabla f\|.$$

We determine first an optimal level $p \in \mathbb{R}$, such that for a given distance d between contours in the domain of f the corresponding change in f is of the form b^p . In other words, we request contours at an equal distance d in the geometric space of the domain, and search for the corresponding level p of contour lines to be used. Substitution gives

$$b^p / d = \|\nabla f\|,$$

hence

$$p = \log(\|\nabla f\|d) / \log b.$$

This value p cannot be used directly, because contours would then depend not only on f , but also on ∇f . Instead, we use p to select two discrete levels L_0 and L_1 :

$$L_i = L(h_i), \text{ with } h_0 = b^{\lfloor p \rfloor} \text{ and } h_1 = b^{\lceil p \rceil}.$$

Furthermore, we use two weights w_0 and w_1

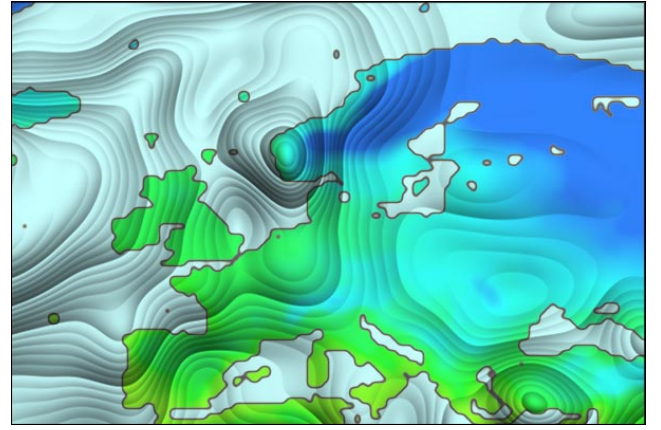
$$w_0 = 1 - (p - \lfloor p \rfloor), \quad w_1 = p - \lfloor p \rfloor.$$

The graphical representation is constructed as a linear combination of graphical elements at the levels L_0 and L_1 , the weights denote the strength to be used per level.

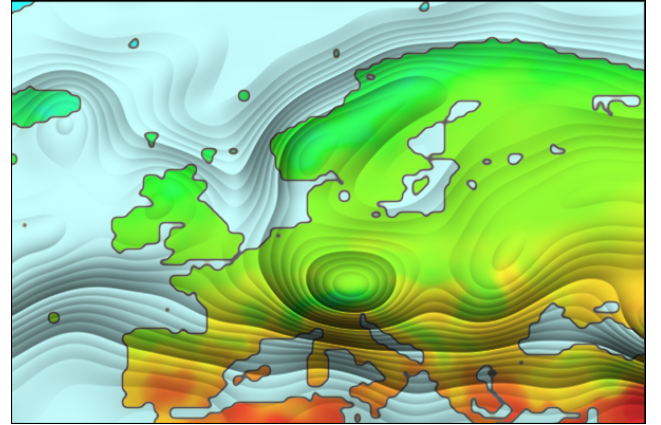
Figure 7 shows a result. The function f is sampled pixel by pixel (similar to [5, 8]) and for each pixel is determined whether or not it lies on a contour. Here we used $b = 2$, i.e. between two diverging contours a single new contour starts. The width and the intensity of the contours are modulated by the weights, leading to tapered ends of contours. Note that the contours of L_1 are a subset of L_0 , hence the sum of the weights for the contours of L_1 equals 1, leading to thick, black lines.

The equally spaced lines lead to a vibrating image, larger structures are difficult to discern, tracking lines visually with the eye is hard. One solution is to use a hierarchy of contours, just like the ticks on a ruler. Instead of a single value for d , we use for instance also $4d$, and use larger weights for the lines with a wide spacing. A result is shown in figure 8.

For a further improvement, we use non-linear mapping to height again. We insert parabola shaped bands again between contours and



January



July

Figure 12: Average temperature and precipitation in Europe

use the weights to blend the parabolas of succeeding levels. Specifically, we use

$$z(\mathbf{x}) = \sum_{i=0}^1 s w_i h_i g \left(\frac{f(\mathbf{x}) \bmod h_i}{h_i} \right),$$

with $g(x) = -ax(1-x)$ again. Figure 9 shows that parabolas between diverging contour lines are split. The visual result, compared to figure 7, is improved, in the sense that the vibration is less. Saddle-points are rendered correctly, though in the shaded version they are harder to interpret.

The strongest effect, however, is obtained when this method is used with parabolas at multiple levels (figure 10). This gives an image that enables the viewer to extract both global, structural information as well as local gradient information. The higher level ridges are shown apart in figure 11. They clearly show how the parabolic ridges smoothly join and split.

5 EXAMPLES

Weather maps are a typical example where the simultaneous display of multiple variables, such as temperature, air pressure, and precipitation, is useful. Examples are shown in figure 12. Data were obtained from the IRI/LDEO Climate Data Library (www.ldeo.columbia.edu). Here the average temperature and precipitation in January and July over the period 1960-1990 are shown. Both for the mapping of the temperature to hue and for the mapping

of precipitation to height a non-linear parabolic mapping was used. These images visualize the different climates in Europe. Another application is the selection of a destination for a summer vacation, looking for a place where temperature is reasonable and precipitation low.

A similar effect could be achieved if only contours instead of ridges would have been used. However, local extrema can much easier here as maxima or minima. Another useful application in the same spirit would be to represent air pressure via height and ridges. Not only low and high pressure areas would be easy to locate, but also the contours are directly meaningful, as they indicate the wind direction.

The resolution of the precipitation data is low, on a grid of 2.5 degrees squared. We also used data from the IPCC Data Distribution Centre (www.ipcc-ddc.cru.uea.ac.uk), which had a resolution of 0.5 degrees squared. A result is shown in figure 13. This image reveals much clearer the strong influence of mountains on precipitation. The use of ridges is less effective for such high-resolution data with high-frequency local features, the locally steep gradients lead to cluttered ridges. A confirmation is shown in figure 14, where we zoomed in on the Alpes, using the same data. Again, smooth structures are visible, well supported by the ridges.

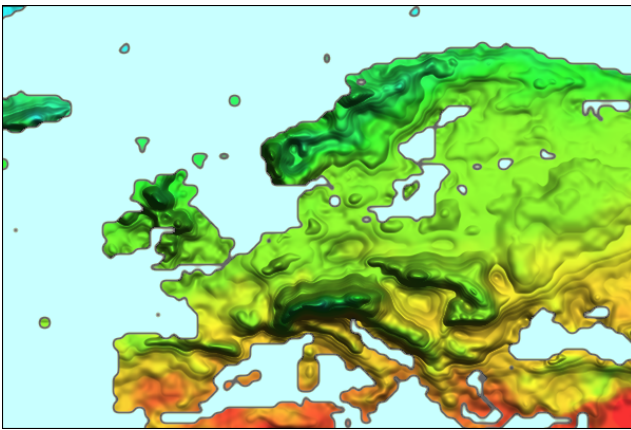


Figure 13: Average temperature and precipitation in Europe, July, high resolution precipitation data

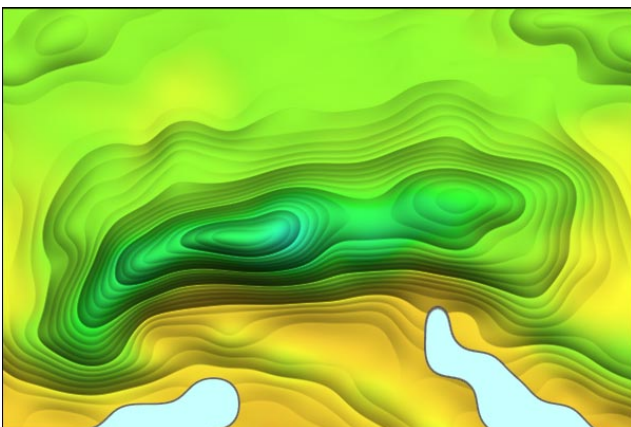


Figure 14: Average temperature and precipitation Alpes, July

6 CONCLUSIONS

We have considered a traditional topic in visualization: the visualization of scalar functions of two variables. We have shown that non-linear mapping of the function, followed by visualization as a shaded height field, can be an effective means to show different aspects of such functions. In other words, we have shown how a visualization of a scalar function of two variables can be enriched by adding ridges. Three types of applications were discussed: visualization of iso-levels, clusters, and dense contours. The method is easy to implement. Furthermore, it is efficient in terms of CPU-time as well as in terms of graphical resources. The images shown here took each at most a few seconds to generate, using non-optimized code on a R5K SGI workstation. Concerning the graphical resources, only grey shades are used, whereas color, texture, and lines can be used for other purposes. And, above all, it leads to intriguing and insightful images.

References

- [1] I. Borg and P. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, New York, 1997.
- [2] W.S. Cleveland. *The Elements of Graphing Data (revised ed.)*. Murray Hill, NJ, 1994.
- [3] B. Jobard and W. Lefer. Creating evenly-spaced streamlines of arbitrary density. In W. Lefer and M. Grave, editors, *Visualization in Scientific Computing '97. Proceedings of the Eurographics Workshop in Boulogne-sur-Mer, France*, pages 43–56, Wien, New York, 1997. Springer Verlag.
- [4] H. Levkowitz and G.T. Herman. Color scales for image data. *IEEE Computer Graphics and Applications*, 12(1):72–80, 1992.
- [5] T. Saito and T. Takahashi. Comprehensible rendering of 3-d shapes. *Computer Graphics*, 24(4):197–206, 1990.
- [6] A. Telea and J.J. van Wijk. Visualization of generalized voronoi diagrams. In *Proceedings Data Visualization 2001*, 2001.
- [7] G. Turk and D. Banks. Image-guided streamline placement. *Computer Graphics*, 30:453–459, 1996.
- [8] J.J. van Wijk. A raster graphics approach to flow visualization. In *Proceedings Eurographics'90*, pages 251–259, Amsterdam, 1990. North-Holland.
- [9] J.J. van Wijk. Implicit stream surfaces. In *Proceedings Visualization '93*, pages 245–252, Los Alamitos, CA, 1993. IEEE Computer Society Press.
- [10] J.J. van Wijk and H. van de Wetering. Cushion treemaps: Visualization of hierarchical information. In *Proceedings IEEE Information Visualization '99*, pages 73 – 78, Los Alamitos, CA, 1999. IEEE Computer Society.
- [11] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2000.
- [12] J.A. Wise, J.J. Thoma, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualising the non-visual: Spatial analysis and interaction with information from text documents. In *Proceedings IEEE Information Visualization '95*, pages 51 – 58, 1995.