VISUALISING ERROR SURFACES FOR ADAPTIVE FILTERS AND OTHER PURPOSES

Mark Fisher, Danilo Mandic, J. Andrew Bangham and Richard Harvey

School of Information Systems, University of East Anglia, Norwich, NR4 7TJ.

ABSTRACT

Modern neural and adaptive systems often have complicated error performance surfaces with many local extrema. Visualising and understanding these surfaces is critical to effective tuning of these systems but almost all visualisation methods are confined to two dimensions. Here we show how to use a morphological scale-space transform to convert these multi-dimensional complex error surfaces into two-dimensional trees where the leaf nodes are local minima and other nodes represent decision points such as saddle points and points of inflection.

1. INTRODUCTION

Visualisation is the process of converting numbers into pictures [1]. The aim is to help understand the underlying phyisical phenomenon. Visualising error surfaces [2] is vital for an effective understanding of many modern signal processing algorithms but is difficult because the error surfaces are often complicated, complex, and defined in more than two dimensions. Attempts to display these surfaces [1] have included plotting two-dimensional functions [3]; contour plots [4, 5]; or density plots, volume rendering, hedgehog plots and tracking critical points [1]. Proposals for reducing the dimensionality of the surface usually amount to projecting the surface into two dimensions – popular strategies are to fix all but two weights or to project into a subspace containing the global optimum of the error performance surface. Unfortunately, as the dimensionality of the underlying adaptive system increases, the number of potential projection planes (defined by a pair of orthogonal axes) increases exponentially. In short, none of these techniques is very effective for education or algorithm design: the plots are hard to reproduce effectively and they represent projections that may not preserve the topology of the surface.

Since the primary interest is the location and characterisation of maxima or minima (extrema) of an error performance surface we propose to adapt an extrema processing technique from scale-space mathematical morphology. The algorithm used here is one from a class known as *sieves* [6, 7] which is related to, but not the same as, alternating sequential filters [8, 9, 10] and greyscale watersheds [11]. The output of a sieve is a tree with leaves that represent local extrema and a structure that depends on the topology of the error surface.

2. BACKGROUND

For convenience assume that the error surface $J(x_1, x_2, \ldots)$ is sampled onto some possibly infinite rectangular grid[1]. If each grid co-ordinate is indexed with a unique integer $v \in V$ where V is a subset of the integers Z, then the cost function may be written as $J(v), v \in V$. Grid co-ordinates that are neighbours may be denoted such by a pair of integers $\{m,n\} = e \in E$. Thus the error surface samples may be defined on a graph $G = (V,E)$ consisting of a set of vertices, V, which are the sample indices and a set of edges E, which are the adjacencies. This notation [8] allows the representation of an N-dimensional image with any specified connectivity. Figure 1 shows an example: a three-dimensional set of 12 samples. If a neighbour is defined as sharing a common side (samples are six-connected) then $V = \{1, \ldots, 12\}$

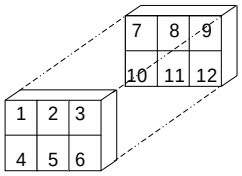

Figure 1: Illustrating a three-dimensional matrix of error samples in which $V = \{1, \ldots, 12\}$, and $E = \{ \{1,7\}, \{1,2\}, \{1,4\}, \{2,8\}, \ldots\}$

For scales, $s \geq 1$, let $C_s(G)$ denote the set of connected subsets of G with s elements. Then with $x \in V$

$$C_s(G,x) = \{\xi \in C_s(G) \mid x \in \xi\}. \quad (1)$$

denotes the set of connected sets of s pixels that contain pixel x. In Figure 1 for example, $C_2(G,5) = \{ \{4,5\}, \{2,5\}, \{5,6\}, \{5,11\} \}$. Equation (1) means that for a point of interest $x \in V$ (usually a maximum or minimum), $C_r(G,x)$ lists all possible r-pixel neighbourhoods of x.

[1] Spatial sampling is not an essential assumption since the technique can be generalised to continuous functions but, in practice, most error surfaces end up sampled.

Equation (1) allows a compact definition of an *opening*, ψ_s, and *closing*, γ_s, of size s, consistent with the proposed notation for graphs and connected sets [8, 6, 7]. The morphological operators, $\psi_s, \gamma_s, M_s, N_s : Z^V \rightarrow Z^V$, may be defined for each $s \geq 1$, as

$$\psi_s J(x) = \max_{\xi \in C_s(G,x)} \min_{u \in \xi} J(u), \quad (2)$$

$$\gamma_s J(x) = \min_{\xi \in C_s(G,x)} \max_{u \in \xi} J(u), \quad (3)$$

and

$$M_s = \gamma_s \psi_s, \qquad N_s = \psi_s \gamma_s. \quad (4)$$

Thus M_s is an opening followed by a closing, both of size s and in any finite dimensional space. The sieves of a function, $J \in Z^V$ are defined in [6] as sequences $(J_s)_{s=1}^{\infty}$ with:

$$J_1 = P_1 J = J, \text{ and } J_{s+1} = P_{s+1} J_s \quad (5)$$

for integers, $s \geq 1$, where P is one of γ, ψ, M or N. Note that, unlike many morphological systems, sieves do not use structuring elements but merge connected sets instead. The algorithm has the effect of locating local extrema in the error surface and "slicing-off" these local peaks and local troughs to produce *flat zones* [10] of s or more samples. Since all the error samples within each extremal connected set have the same value, a simple graph reduction at each stage can lead to a fast algorithm [9]. At subsequent scales, larger extrema are removed, so the processor formally satisfies the scale-space causality requirements [12, 13]. The differences between successive outputs

$$d^s = J_s - J_{s-1} \quad (6)$$

are called *granule functions* and non-zero regions within d^s are called *granules* denoted by d_j^s where $j = 1, \ldots, N(s)$ indexes the number of granules, $N(s)$, at scale s. As scale s increases, $N(s)$ decreases, since the granules are larger. At the final scale a tree, $T = (N, A)$ may be built using the output of a sieve $(d_s)_{s=1}^{S}$ which is also a graph with vertices, or nodes N, and edges A. The tree has the following properties:

1. If the sampled error surface has S samples then the root of the tree, $R(T)$ maps to d_1^S which is the whole surface.

2. If $a \in A$ with $a = (n_p, n_c)$ then n_c is a child of n_p and $d_{n_c}^{s_c} \subset d_{n_c}^{s_p}$.

In other words, because the sieve is removing local extrema, granules at some scale s_c are always contained within granules at some greater scale, s_p, unless $s_c = S$ in which case it is the root. The tree encodes the containment of granules, equipotential zones, within the error surface. M- and N-sieves encode the positions of maxima and minima simultaneously which for image processing is useful since it is often postulated that local maxima and minima are objects [12], but for error surface visualisation one tends to be interested in either the minima or maxima in which case either the opening or closing sieve is appropriate.

3. TREE-BASED VISUALISATION

Although the mathematics of these sieves is intricate [6], it is not complicated to explain these processors through an example. Figure 2 (top) shows an example of a well known benchmark error surface (the Himmelblau function [14, 15])

$$E(x_1, x_2) = \left(x_1^2 + x_2 - 11\right)^2 + \left(x_1 + x_2^2 - 7\right)^2 \quad (7)$$

that has four minima. Such surfaces are not uncommon in signal processing (see [16] for an example of a neural network with four maxima) but as in Figure 2 they may not be easy to visualise. Here, for example, one of the minima has disappeared behind another. The lower part of Figure 2

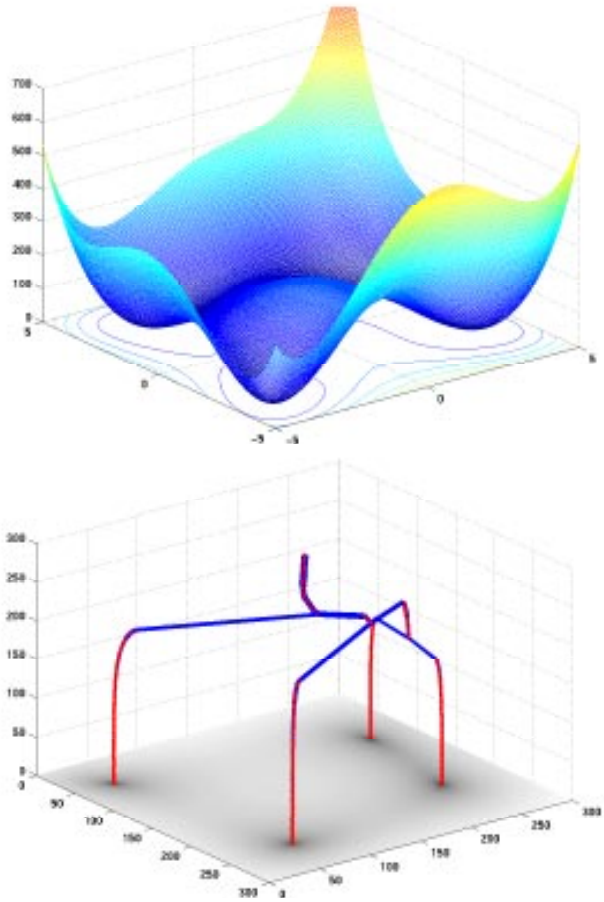

Figure 2: The Himmelblau function visualised (top) as a conventional surface plotted against x_1 and x_2 and (bottom) as a closing scale tree with the image plotted at $z = 0$

shows the same function visualised as a closing tree with a

greyscale representation of the function on the $z = 0$ axis. The tree is shown as dots representing the granules connected with lines showing containment. Each dot is plotted with x, y co-ordinates equal to the centroid of its corresponding granule. The z-axis has been used to plot scale.

The root of the tree contains all samples of the error surface since it represents the interior of a contour $J(v) < \infty$. Successive operations of the sieve give granules that are connected regions corresponding to the interiors of equipotential contours drawn around local minima. In an analogy with watersheds one can imagine the tree being built root-first by the error surface filled with water draining through the local minima. The tree bifurcates where the single sheet of water becomes two separate pools with a watershed between them. The process terminates when all the water has drained through the local minima. We emphasize that the technique described here is more general than watersheds because it can be defined in any finite dimensional space and can process maxima, minima or extrema.

Figure 2 shows that the tree captures many aspects of the surface. There are four local minima which is not clear from the top of Figure 2 and that the left-hand minima has a larger domain of attraction than the other three. Comparing this to Figure 3 which shows a complex benchmark function

$$f(z) = \left|z^3 - 1\right|^2, \text{ where } z = x_1 + jx_2 \tag{8}$$

that has symmetric minima shows that if the three basins of attraction have identical geometry the tree will split three ways.

Furthermore the tree is a data structure, so it is simple to store information about an optimiser at the nodes and vice versa – the optimiser's trajectory can be described as a trajectory through V.

4. DISCUSSION

The scale trees described here are a useful tool for visualising error performance surfaces. Their leaves represent local minima and their branch structure indicates the topology of the surface.

A slight complication arises if the error surface is noisy, as in Figure 4 (centre). Such surfaces are commonplace in real signal processing evaluations and the overall effect is to introduce small-scale perturbations in the surface. The underlying structure of the resulting tree is still visible but a pragmatic approach is suggested by a series of experiments [17] that show that the sieve is almost as effective at noise removal as a matched filter. Here an M-sieve to scale 10 has been applied to remove noise. The result is shown on the right of Figure 4. Much of the complicated detail has been removed, leaving the important structure. A more subtle approach, similar to wavelet denoising, is to remove child nodes that are not significantly different from their parents are removed.

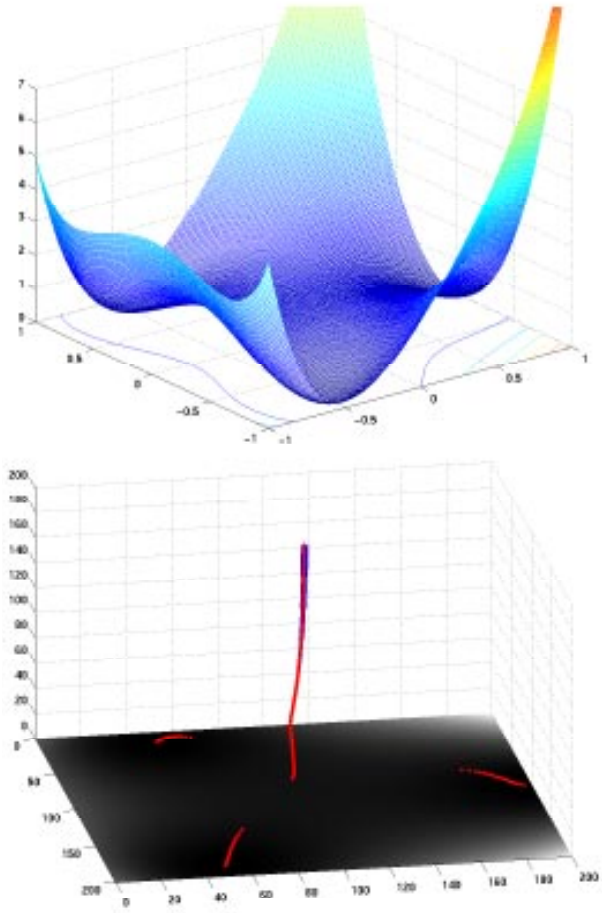

Figure 3: $F(z) = \left|z^3 - 1\right|^2$ visualised (top) as a conventional surface and (bottom) as an closing scale tree.

5. SUMMARY

A novel method for visualising error performance surfaces of adaptive algorithms has been provided. It uses a tree structure that comes from the sieve representation of images of error performance surfaces. Each node represents a local equipotential contour and hence many configurations of the underlying system, but the output of the sieve is unique given a particular image and this output and the image form an invertible transform.

This approach therefore gives the character, position, and basin of attraction of minima in the error performance surface via an algorithm that has low order complexity. Although, for simplicity we study here only the 2D benchmark problems studied the by authors, the technique is defined for any finite-dimensional surface so this study paves the way for analysis of multidimensional and complex error perfor-

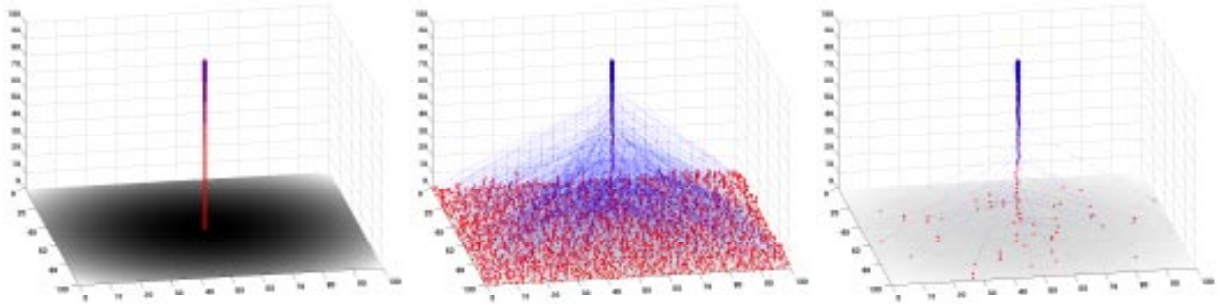

Figure 4: A quadratic error surface and its tree (left); a noisy quadratic error surface (centre) and the tree after removing all local extrema of area 10 or less (right)

mance surfaces which at present have to be analysed by projections onto supporting planes.

6. REFERENCES

[1] D. Silver and N. J. Zabusky, "Scientific visualization and computer vision," in *Proc. IEEE Workshop on Visualization and Machine Vision*, pp. 55–61, 1994.

[2] R. J. Moorhead and Z. Zhu, "Signal processing aspects of scientific visualization," *IEEE Signal Processing Magazine*, vol. 12, no. 5, pp. 20–41, 1995.

[3] D. R. Hush, B. Horne, and J. M. Salas, "Error surfaces for multilayer perceptrons," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 22, no. 5, pp. 1152–1161, 1992.

[4] M. Nayeri and W. K. Jenkins, "Alternate realizations to adaptive IIR filters and properties of their performance surfaces," *IEEE Trans. Circuits and Systems*, vol. 36, no. 4, pp. 485–496, 1989.

[5] S. D. Stearns, "Error surfaces of recursive adaptive filters," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. ASSP-29, no. 3, pp. 763–766, 1981.

[6] J. A. Bangham, R. Harvey, and P. D. Ling, "Morphological scale-space preserving transforms in many dimensions," *J. Electronic Imaging*, vol. 5, no. 3, pp. 283–299, 1996.

[7] J. A. Bangham, P. W. Ling, and R. Harvey, "Nonlinear scale-space causality preserving filters," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, pp. 520–528, 1996.

[8] L. Vincent, "Graphs and mathematical morphology," *Signal Processing*, vol. 16, pp. 365–388, 1989.

[9] L. Vincent, "Morphological grayscale reconstruction in image analysis: applications and efficient algorithms," *IEEE Trans. Image Processing*, vol. 2, no. 2, pp. 176–201, 1993.

[10] P. Salembier and J. Serra, "Flat zones filtering, connected operators and filters by reconstruction," *IEEE Trans. Image Processing*, vol. 8, no. 4, pp. 1153–1160, 1995.

[11] M. Fisher and R. Aldridge, "Hierarchical image segmentation using a watershed scale-space tree," in *Proc. Seventh International Conference on Image Processing and its applications*, vol. 2, pp. 522–526, 1999.

[12] T. Lindeberg, *Scale-space theory in computer vision*. Kluwer, 1994.

[13] R. Harvey, J. A. Bangham, and A. Bosson, "Scale-space filters and their robustness," in *Scale-space theory in computer vision* (B. ter Haar Romeny, L. Florack, J. Koenderink, and M. Viergever, eds.), pp. 341–344, Spinger, 1997.

[14] G. S. Androulakis, M. N. Vrahatis, and T. Grapsa, "Studying the performance of optimization methods by visualization," *SAMS*, vol. 25, pp. 21–42, 1996.

[15] C. A. Botsaris, "A curvilinear optimization method based upon iterative estimation of the eigensystem of the Hessian matrix," *J. Math. Anal. Appl.*, vol. 63, pp. 396–411, 1978.

[16] J. Y. Han, M. R. Sayeh, and J. Zhang, "Convergence and limit points of neural network and its application to pattern recognition," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1217–1222, 1989.

[17] R. Harvey, A. Bosson, and J. A. Bangham, "A comparison of linear and nonlinear scale-space filters in noise," in *Proc. Eusipco'96*, vol. III, pp. 1777–1780, 1996.