1 Horizon Enhancing Volume Rendering

The obvious approach to highlight extrema would be computing the derivative, e.g. with central differences to find the extrema and then modulate the result so that small values would map to high opacities and vice versa. A natural fit for such a mapping would be a smooth bump function ϕ:

$$\phi(v) = \begin{cases} \exp^{-\frac{1}{1-v^2}} & \text{for } |v| < 1 \\ 0 & \text{else} \end{cases} \quad (1)$$

where v is the derivative $f' = df(x,y,z)/dz$ scaled with a user defined scale factor α to define the threshold for mapping the result to 0. The scaling factor o_ϕ for the opacity at position (x,y,z) can then be defined by

$$o_\phi(x,y,z) = \phi(\alpha \cdot f'(x,y,z)). \quad (2)$$

The result of such an operation on a seismic trace can be seen in Figure 1c. The magenta curve represents the function values of the seismic trace, the cyan one the resulting opacity scale factor. As can be seen in the figure, monotonous rising or falling parts in the input signal are nicely filtered out of the result as desired. The approach, however, has some drawbacks. First of all, like local extrema, the first derivative of saddle-points maps to zero, resulting in over-classification of these areas (compare green box in Figure 1c). In addition the threshold parameter α has to be defined by the user and does not necessarily work equally well over the whole dataset. Compare the purple boxes in Figure 1c. The left box shows an area of over-classification. The gradient here is below the threshold and thus the otherwise monotonously falling part is defined as an extremum. The right purple box shows an extremum which is nearly missed, because of the sharp edges between sampling points. Instead of the previously described approach we decided to use a non-linear technique based on comparison of the local neighborhood. We compare pairs of symmetrical samples in front and behind the current sample to the current sample itself. If the difference between the current sample and both neighbor samples has the same sign there must be a local extremum between the neighbor samples. In the following we explain our approach for local maxima, only. Minima can be found with the same technique by simply inverting the comparison operations. For a formal definition see Equation 3. By just sampling the direct neighbors with $n = 1$ Equation 3 returns 1, if the current sample is a local maximum and 0 if not.

$$\varphi(x,y,z,n) = \begin{cases} 1 & \text{for } f(x,y,z-n) < f(x,y,z) \wedge f(x,y,z+n) < f(x,y,z) \\ 0 & \text{else} \end{cases} \quad (3)$$

By expanding the neighborhood the distance of the current sample to the maximum can be estimated. If the result of Equation 3 is 1 for $n = 2$, but 0 for $n = 1$ there exists a maximum between $z-2$ and $z+2$ but not at z. Thus the distance from the center must be 1. Of course maxima can only be found in the neighborhood used for sampling, thus the maximal detectable distance is one half of the neighborhood size. We can use that to define a scale factor o_φ using a local neighborhood of size $k \cdot 2$ for the opacity by

$$o_\varphi(x,y,z,n) = \frac{1}{k}\sum_{n=1}^{k} \varphi(x,y,z,n) \quad (4)$$

The plot in Figure 1b shows the same trace mentioned before, but here we applied Equation 4 in combination with its counterpart to detect minima with $k = 3$. The over-classification errors resulting from Equation 2 are not an issue here and detection of the peaks is more stable. The saddle point, which should not be detected, as we specifically look for minima and maxima is assigned some opacity (green box in Figure 1b). This is probably caused by noise. By simply summing up the results from all pairs single outlier voxels will contribute to the sum when computing φ_1. To reduce this problem we apply some implicit filtering, Instead of just computing the sum over all φ_n we define a minimum size s for the peak:

$$\varphi_s(x,y,z,n) = \begin{cases} \varphi(x,y,z,n) & \text{for } (n \geq s) \vee (n < s \wedge \varphi(x,y,z,n+1) = 1) \\ 0 & \text{else.} \end{cases} \quad (5)$$

The scale factor for the opacity is then computed using φ_s in Equation 4 instead of φ. Figure 1a shows the the technique applied to the same trace as before with $k = 1$ and $s = 3$. For rendering we compute the scale factor on the fly in the raycasting fragment shader and apply it the same way as the gradient magnitude modulated shading.

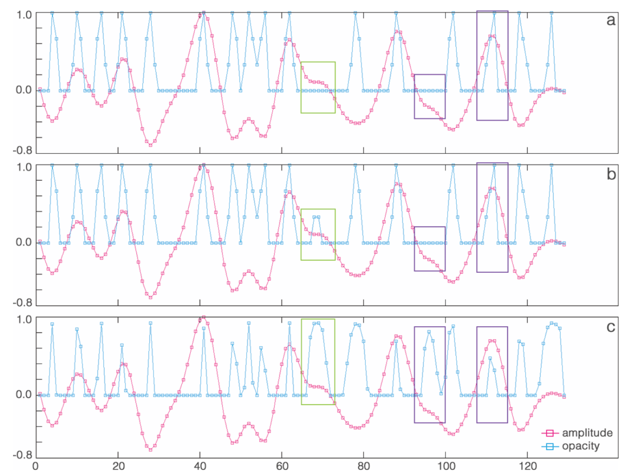

Fig. 1: The different opacity modulations applied to a single trace. The amplitude, scaled to $[-1.0..1.0]$ and resulting opacity modulation factor is plotted on the x-axis, the timestep in the trace on the y-axis. Figures a shows the implemented approach, with a minimum peak size $s = 3$, Figure b without a minimum peak size. $k = 3$ for a and b. Figure c shows the derivative modulated by a smooth bump function. Some critical differences are highlighted (compare text).

References