Image-guided blended neighbor interpolation of scattered data

Dave Hale, Center for Wave Phenomena, Colorado School of Mines

SUMMARY

Uniformly sampled images are often used to interpolate other data acquired more sparsely with an entirely different mode of measurement. For example, downhole tools enable geophysical properties to be measured with high precision near boreholes that are scattered spatially, and less precise seismic images acquired at the earth's surface are used to interpolate those properties at locations far away from the boreholes. *Image-guided interpolation* is designed specifically to enhance this process.

Most existing methods for interpolation require distances from points where data will be interpolated to nearby points where data are known. Image-guided interpolation requires non-Euclidean distances in metric tensor fields that represent the coherence, orientations and shapes of features in images. This requirement leads to a new method for interpolating scattered data that I call *blended neighbor interpolation*. For simple Euclidean distances, blended neighbor interpolation resembles the classic natural neighbor interpolation.

INTRODUCTION

Interpolation of spatially scattered data is a common task in geophysics. Assume that we have obtained a set $\mathscr{F} = \{f_1, f_2, \ldots, f_K\}$ of K known sample values $f_k \in \mathbb{R}$ that correspond to a set $\mathscr{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K\}$ of K known sample points $\mathbf{x}_k \in \mathbb{R}^n$. Together these two sets comprise a set

$$\mathscr{K} = \{(f_1, \mathbf{x}_1), (f_2, \mathbf{x}_2), \ldots, (f_K, \mathbf{x}_K)\} \tag{1}$$

of K known samples. These samples are scattered in the sense that the n-dimensional sample points in the set $\mathscr{X}$ may have no regular geometric structure. The interpolation problem is to use the known samples in $\mathscr{K}$ to construct a function $q(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$, such that $q(\mathbf{x}_k) = f_k$.

This problem has no unique solution, as there exist an infinite number of functions $q(\mathbf{x})$ that satisfy the interpolation conditions $q(\mathbf{x}_k) = f_k$. We therefore may also require that the interpolant $q(\mathbf{x})$ be smooth, bounded, and fast to compute. Because tradeoffs exist among such criteria, a variety of methods for interpolating scattered data are commonly used today. Foster and Evans (2008) provide a recent evaluation of several methods in the context of a geophysical application.

Most interpolation methods define (perhaps implicitly) each value of the interpolant $q(\mathbf{x})$ as a weighted sum of known sample values f_k, where the weights depend in various ways on the Euclidian distances $\|\mathbf{x} - \mathbf{x}_k\|$ between the point $\mathbf{x}$ and known sample points $\mathbf{x}_k$. The simplest example is the nearest neighbor interpolant, which sets $q(\mathbf{x})$ equal to the value f_k corresponding to the sample point $\mathbf{x}_k$ nearest to the interpolation point $\mathbf{x}$.

In image-guided interpolation, I replace distance with time or, more precisely, a non-Euclidean distance defined for a metric tensor field.

Metric tensor fields

I compute a metric tensor field $\mathbf{D}(\mathbf{x})$ from structure tensors $\mathbf{S}(\mathbf{x})$, which are smoothed outer products of image gradients (van Vliet and Verbeek, 1995). Figure 1 illustrates a tensor field $\mathbf{D}(\mathbf{x})$ computed for a seismic image. Intuitively, tensors in $\mathbf{D}(\mathbf{x})$ alter the simple constant-velocity time-distance relationship so that known samples within a geologic layer are nearer (in time) than are samples in different layers or in locations or directions where coherence is low.

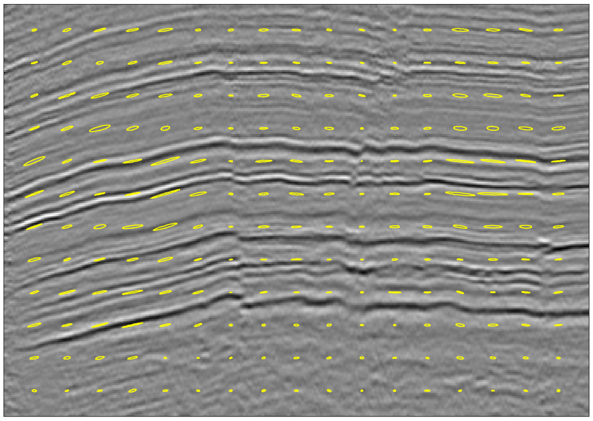


Figure 1: A tensor field $\mathbf{D}(\mathbf{x})$ computed for a 2D seismic image. Ellipses here represent only a small subset of the tensors computed for every image sample; they indicate the coherence, orientations and linearity of image features.

In this example, I computed the tensor field $\mathbf{D}(\mathbf{x}) = \mathbf{S}^{-1}(\mathbf{x})/[1 - c(\mathbf{x})]$, where the function $c(\mathbf{x})$ is a measure of coherence, $0 \le c(\mathbf{x}) < 1$, here computed from structure tensors $\mathbf{S}(\mathbf{x})$ using the method suggested by Fehmers and Höcker (2003). Locations and directions of high coherence within layers correspond to elongated ellipses in Figure 1.

IMAGE-GUIDED BLENDED NEIGHBOR INTERPOLATION

Given a tensor field $\mathbf{D}(\mathbf{x})$, image-guided blended neighbor interpolation is a two-step process:

Step 1: solve the eikonal equation

$$\begin{aligned} \nabla t(\mathbf{x}) \cdot \mathbf{D}(\mathbf{x}) \, \nabla t(\mathbf{x}) &= 1, \quad \mathbf{x} \notin \mathscr{X}; \\ t(\mathbf{x}) &= 0, \quad \mathbf{x} \in \mathscr{X} \end{aligned} \tag{2}$$

for

$t(\mathbf{x})$**:** the minimum traveltime from $\mathbf{x}$ to the nearest (in time) known sample point $\mathbf{x}_k$, and

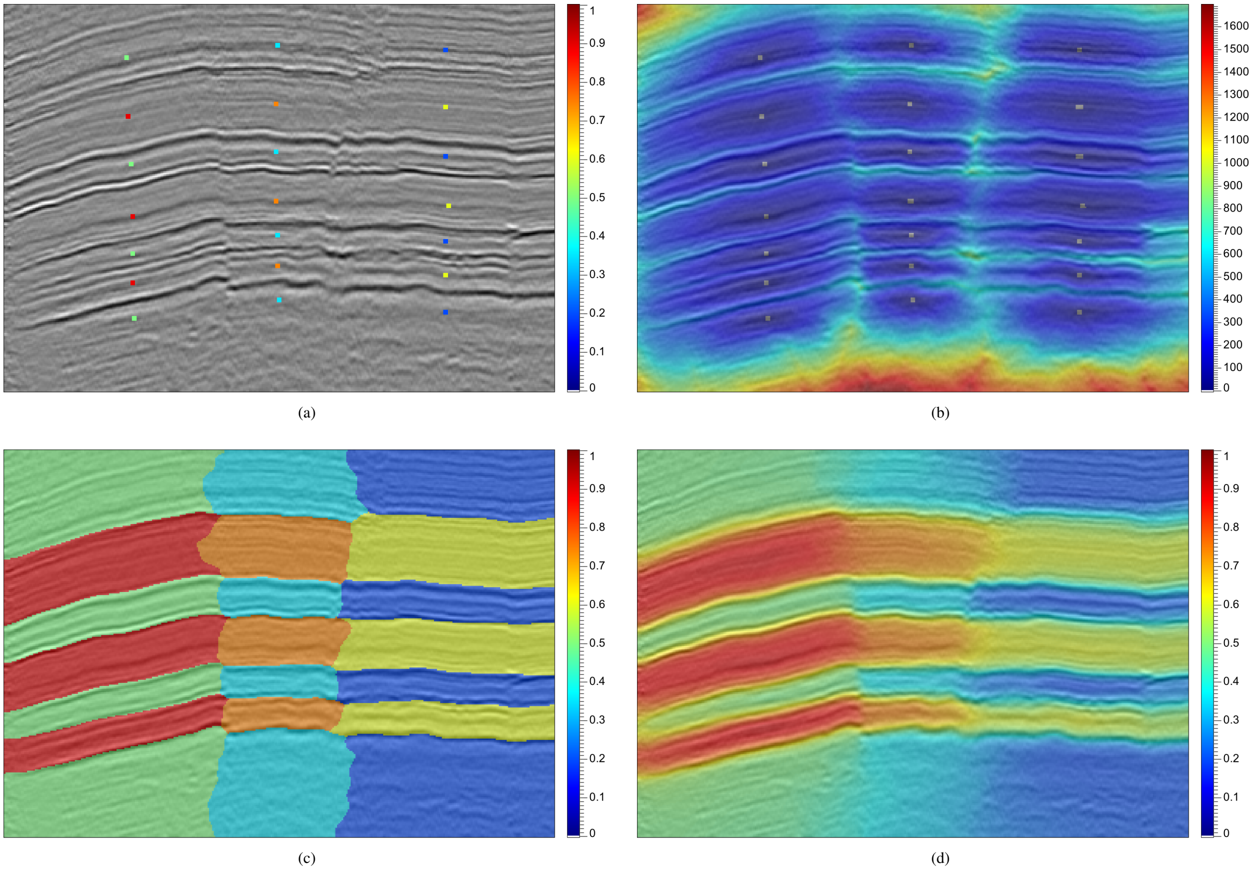
$p(\mathbf{x})$**:** the value f_k corresponding to the sample point $\mathbf{x}_k$ nearest (in time) to the point $\mathbf{x}$.

Step 2: solve the smoothing equation

$$q(\mathbf{x}) - \frac{1}{2} \nabla \cdot t^2(\mathbf{x}) \mathbf{D}(\mathbf{x}) \, \nabla q(\mathbf{x}) = p(\mathbf{x}), \tag{3}$$

for the interpolant $q(\mathbf{x})$.

Figure 2 illustrates this process for the tensor field in Figure 1. In this example, 21 scattered samples in Figure 2a have colors corresponding to values that alternate vertically while decreasing from left to right. I painted these samples interactively using a digital 3×3-pixel paintbrush to make the sample points more clearly visible than if only one pixel per sample were painted. Such interactive painting can be use-



(a)

(b)

(c)

(d)

Figure 2: Image-guided blended neighbor interpolation. For a set $\mathscr{K}$ of known samples (a), we first solve equation 2 to compute the time map $t(\mathbf{x})$ (b) and nearest neighbor interpolant $p(\mathbf{x})$ (c), and then solve equation 3 for the blended neighbor interpolant $q(\mathbf{x})$ (d).

ful in geologic interpretation of seismic images, either with or without additional scattered data obtained from borehole measurements.

In step 1, I computed the time map $t(\mathbf{x})$ shown in Figure 2b by solving a finite-difference approximation to the eikonal equation 2. I used an iterative algorithm similar to that proposed by Jeong et al. (2007). Other suitable algorithms include variants of fast-marching methods (Sethian, 1999; Sethian and Vladimirsky, 2005; Konukoglu et al., 2007) and sweeping methods (Tsai et al., 2003; Qian et al., 2007), but care must be taken to handle correctly the anisotropic and inhomogeneous coefficients $\mathbf{D}(\mathbf{x})$ in this equation. Ridges in the time map $t(\mathbf{x})$ are aligned with near-vertical faults in the seismic image, because times increase rapidly where elements of the tensor field $\mathbf{D}(\mathbf{x})$ (the ellipses displayed in Figure 1) are small.

While computing the time map $t(\mathbf{x})$, I also computed the nearest neighbor interpolant $p(\mathbf{x})$ displayed in Figure 2c. It may be possible to compute $p(\mathbf{x})$ directly from $t(\mathbf{x})$, but I have not found an efficient and stable way to do so. Instead I compute $p(\mathbf{x})$ as I solve for times $t(\mathbf{x})$.

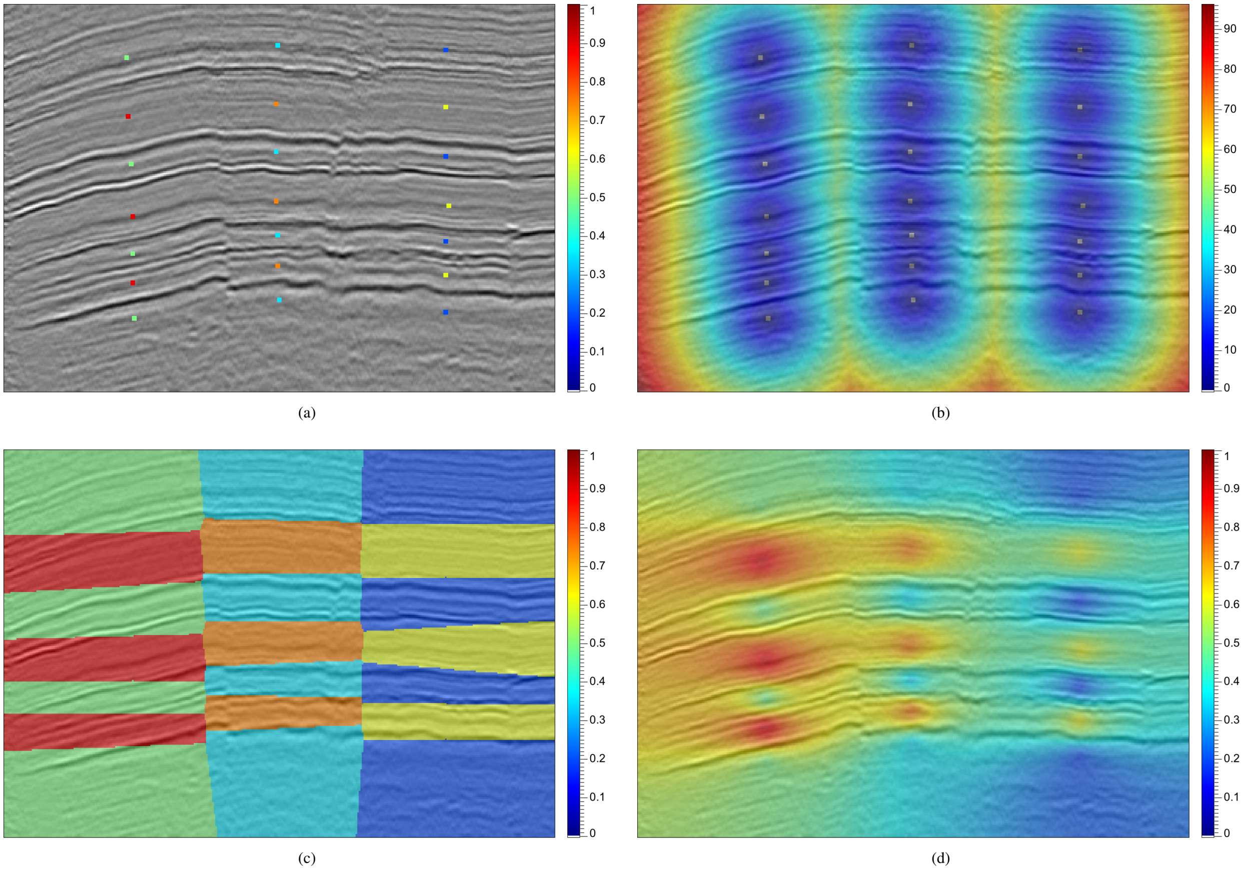
Step 2 requires the solution of another partial differential equation 3 for the desired interpolant $q(\mathbf{x})$. Solution of this equation is equivalent to smoothing the nearest neighbor interpolant $p(\mathbf{x})$ computed in step 1. In this sense, equation 3 blends nearest neighbor values, so I call the complete two-step process *image-guided blended neighbor interpolation*.

Note that smoothing or blending does not alter the values of known samples. At known sample points $\mathbf{x}_k$ for which $t(\mathbf{x}_k) = 0$, the solution to equation 3 is clearly $q(\mathbf{x}_k) = p(\mathbf{x}_k) = f_k$; the interpolation conditions are satisfied.

In the example of Figure 2, I solved a finite-difference approximation to the blending equation 3 using an iterative conjugate-gradient algorithm. With $K = 21$ scattered known samples and $N = 251 \times 357$ image samples, I performed 709 iterations to converge to the blended neighbor interpolant $q(\mathbf{x})$ displayed in Figure 2d. More generally, the number of iterations is proportional to the largest times $t(\mathbf{x})$ computed in step 1.

The blended neighbor interpolant in Figure 2d conforms to features in the seismic image. Contours of constant color are aligned with both near-horizontal features (geologic layers) and near-vertical discontinuities (geologic faults).

For comparison, Figure 3 shows the result of blended neighbor interpolation for an isotropic constant tensor field $\mathbf{D}(\mathbf{x}) = \mathbf{I}$, where $\mathbf{I}$ denotes the identity matrix. In this *image-ignorant* interpolation, time $t(\mathbf{x})$ computed in step 1 is simply Euclidean distance. The blended neighbor interpolant $q(\mathbf{x})$ shown in Figure 3d is therefore smooth, but does not conform to image features. I used 243 conjugate-gradient iterations in step 2 to converge to this interpolant.



(a)

(b)

(c)

(d)

Figure 3: Image-ignorant blended neighbor interpolation, for $\mathbf{D}(\mathbf{x}) = \mathbf{I}$. For a set $\mathscr{K}$ of known samples (a), we first use equation 2 to compute the time (here, Euclidean distance) map (b) and nearest neighbor interpolant (c), and then solve equation 3 for the blended neighbor interpolant (d).

ALTERNATIVE INTERPOLATION METHODS

Table 1 lists several interpolation methods along with attributes that best distinguish them. In developing blended neighbor interpolation, I considered all of these methods, and rejected methods (including some not listed) that would be more difficult to extend to efficient image-guided interpolation in arbitrary tensor fields $\mathbf{D}(\mathbf{x})$. Figure 4 illustrates four of these interpolants in the isotropic constant case, $\mathbf{D}(\mathbf{x}) = \mathbf{I}$.

Among these methods, harmonic and biharmonic interpolation (e.g., Mitášová and Luboš, 1993) are easiest to extend to interpolation in tensor fields $\mathbf{D}(\mathbf{x})$, because they require no explicit computation of distances or times. Unfortunately, as illustrated in Figure 4a, the harmonic interpolant has sharp cusps at the known sample points $\mathbf{x}_k$.

Method	Smooth	Bounded	Image-guided
Harmonic	no	yes	fast
Biharmonic	yes	no	slow
Nearest neighbor	no	yes	fast
Natural neighbor	fair	yes	slow
Blended neighbor	**fair**	**yes**	**fast**

Table 1: Interpolation methods briefly compared. Examples for all but the biharmonic method are shown in Figure 4.

The biharmonic interpolant (not shown) is much smoother, but in contrast to the interpolants shown in Figure 4, interpolated values $q(\mathbf{x})$ are not bounded by known sample values f_k. Furthermore, biharmonic interpolation is a relatively slow method in the image-guided case of anisotropic and varying tensor fields $\mathbf{D}(\mathbf{x})$.

Nearest neighbor interpolation is relatively fast in that case, but the interpolant is discontinuous, as shown in Figure 4b. Among all methods listed in Table 1, nearest neighbor interpolation is the only one that lacks linear precision; that is, it fails to interpolate precisely samples $(f_k, \mathbf{x}_k)$ of any linear function $f(\mathbf{x})$.

The natural neighbor interpolant (Sibson, 1981; Watson and Phillip, 1987; Sambridge et al., 1995) is most relevant here because, like the blended neighbor interpolant, it can be computed by smoothing the nearest neighbor interpolant. I computed the natural neighbor interpolant shown in Figure 4c using the discrete smoothing method of Park et al. (2006). That smoothing method becomes relatively slow (as for biharmonic interpolation) if extended to image-guided interpolation in tensor fields $\mathbf{D}(\mathbf{x})$.

Indeed, this loss of efficiency inspired the more efficient smoothing implied by solution of the blending equation 3. Blended neighbor interpolation is most like discrete natural neighbor interpolation, but with a different and faster smoothing method. It is therefore not surprising that the blended neighbor interpolant in Figure 4d is similar to the natural neighbor interpolant in Figure 4c.

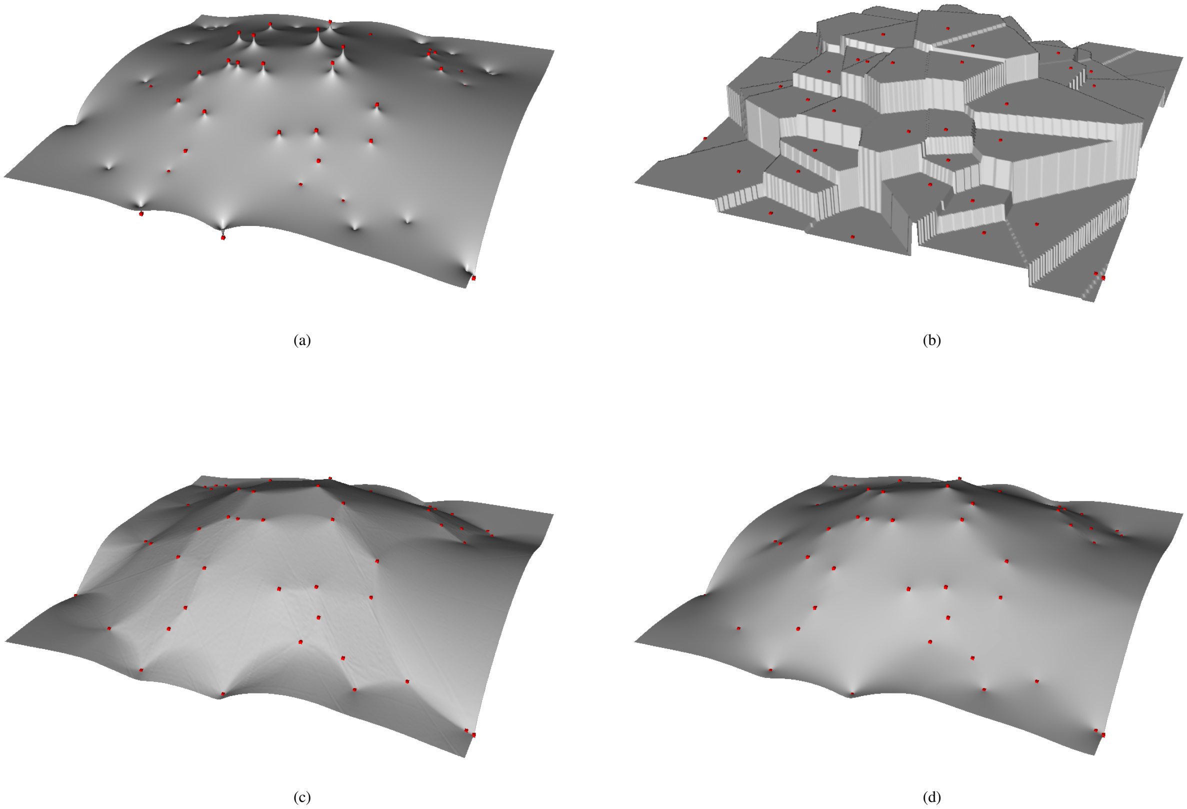


Figure 4: Four different interpolants — (a) harmonic, (b) nearest neighbor, (c) natural neighbor and (d) blended neighbor — of scattered data.

DISCUSSION AND CONCLUSION

When an image is available, the accuracy of image-guided interpolation depends on the extent to which the property being interpolated is correlated with image features. If no such correlation exists, then a simpler and faster image-ignorant interpolation may be more accurate than image-guided interpolation with an irrelevant image.

In many contexts, however, an isotropic and constant tensor field is inappropriate. And even when a useful image is unavailable, it may still be possible to construct a more appropriate tensor field $\mathbf{D}(\mathbf{x})$ to guide interpolation. The two-step process proposed here could more precisely be called tensor-guided blended neighbor interpolation, because we actually use only the tensor field $\mathbf{D}(\mathbf{x})$, not the image.

However, in the example shown in this paper, I derived that tensor field from a uniformly-sampled seismic image, and then interpolated scattered data on the same uniform sampling grid. In some applications, it may be desirable to interpolate scattered data with higher resolution, and nothing in the method prevents this. Image-guided blended neighbor interpolation requires only that we provide a tensor $\mathbf{D}$ for all uniformly sampled locations $\mathbf{x}$ where we interpolate.

Tensor fields used in image-guided blended neighbor interpolation are analogous to spatial correlation functions (variograms) used in kriging, a geostatistical interpolation method in which subsurface properties are modelled as random variables (e.g., Goovaerts, 1997). But the two interpolation methods are otherwise rather different. Where images have less resolution than desired for geostatistical modeling, blended neighbor interpolation may be used to provide image-guided trends for kriging and other geostatistical methods.

Blended neighbor interpolation of scattered data is most similar to the classic method of natural neighbor interpolation, in that both methods smooth a nearest neighbor interpolant, and the extent of smoothing grows with distance to the nearest known sample point. Smoothing in blended neighbor interpolation is implied by the solution of a partial differential equation, whereas smoothing in natural neighbor interpolation is performed by explicitly computed weighted sums of nearest neighbor sample values.

Compared with horizon-based methods commonly used today when interpreting seismic images, image-guided blended neighbor interpolation represents a new and more direct way to use such images to interpolate scattered data. While the examples shown in this paper are 2D images, this process was designed to work as well for 3D images.

ACKNOWLEDGMENTS

Thanks to Luming Liang for many thoughtful discussions of the ideas presented in this paper, and to the U.S. Department of Energy for providing the seismic image shown in the examples.

REFERENCES

Fehmers, G.C., C.F.W. Höcker, 2003, Fast structural interpretations with structure-oriented filtering: Geophysics, **68**, 1286–1293.

Foster, M.P., and A.N. Evans, 2008, An evaluation of interpolation techniques for reconstructing ionospheric TEC maps: IEEE Transactions on Geoscience and Remote Sensing, **46**, 2153–2164.

Goovaerts, P., 1997, Geostatistics for natural resources evaluation: Oxford University Press.

Jeong, W.-K., P.T. Fletcher, R. Tao and R.T. Whitaker, 2007, Interactive visualization of volumetric white matter connectivity in DT-MRI using a parallel-hardware Hamilton-Jacobi solver: IEEE Transactions on Visualization and Computer Graphics, **13**, 1480–1487.

Konukoglu, E., M. Sermesant, O. Clatz, J.-M. Perat, H. Delingette and N. Ayache, 2007, A recursive anisotropic fast marching approach to reaction diffusion equation: application to tumor growth modeling: Information processing in medical imaging, **20**, 687–699.

Mitášová, H. and M. Luboš, 1993, Interpolation by regularized spline with tension: I. theory and implementation: Mathematical Geology, **25**, 641–655.

Park, S.W., L. Linsen, O. Kreylos, J.D. Owens, B. Hamann, 2006, Discrete Sibson interpolation: IEEE Transactions on Visualization and Computer Graphics, **12**, 243–253.

Qian, J., Y.-T.. Zhang, and H.-K. Zhao, 2007, A fast sweeping method for static convex Hamilton-Jacobi equations: Journal of Scientific Computing, **31**, 237–271.

Sambridge, M., J. Braun and H. McQueen, 1995, Geophysical parameterization and interpolation of irregular data using natural neighbors: Geophysical Journal International, **122**, 837–857.

Sethian, J.A, 1999, Fast marching methods: SIAM Review, **41**, 199–235.

Sethian, J.A, and A. Vladimirsky, 2005, Ordered upwind methods for static Hamilton-Jacobi equations: theory and algorithms: SIAM Journal of Numerical Analysis, **41**, 325–363.

Sibson, R., 1981, A brief description of natural neighbor interpolation, *in* V. Barnett, ed., Interpreting Multivariate Data: John Wiley & Sons, 21–36.

Tsai, Y.-H.R., L.-T. Cheng, S. Osher and H.-K. Zhao, 2003, Fast sweeping algorithms for a class of Hamilton-Jacobi equations: SIAM Journal of Numerical Analysis, **41**, 673–694.

van Vliet, L.J., and P.W. Verbeek, 1995, Estimators for orientation and anisotropy in digitized images: Proceedings of the first annual conference of the Advanced School for Computing and Imaging ASCI'95, Heijen (The Netherlands), 442–450.

Watson, D.F. and G.M. Phillip, 1987, Neighbor-based interpolation: Geobyte, **2**, 12–16.