# Seeing and Hearing Dynamic Loess Surfaces

W. M. Coughran, Jr.
*AT&T Bell Laboratories*
*Murray Hill, NJ 07974, USA.*
wmc@research.att.com

Eric Grosse
*AT&T Bell Laboratories*
*Murray Hill, NJ 07974, USA.*
ehg@research.att.com

## Abstract

Scientific video, combining animated images with sound, is a powerful tool for understanding transient two-dimensional or static three-dimensional data. Animation of colored perspective plots often reveals subtle data characteristics not seen in a series of static images. Using the multidimensional data fitting technique *loess*, it is possible to construct dashed plots that visually represent the smoothed approximation and its local error. Sound can be used to add scalar parameters like time "tick marks" or the amplitude of an associated quantity; we describe means of processing such scalar data using variation-diminishing splines and specifics of sound generation including loudness equilibration. We also explain the limitations of our techniques and suggest some extensions.

## 1 Introduction

Our long-term research interests have been in simulation techniques and data fitting. Measured or simulated data often occur as values sampled at scattered locations in time and two or three spatial variables. Moreover, there are often one or more scalar parameters associated with such data sets, representing time, a global error value, an integral, and so forth. Such complex data fields are difficult to comprehend, but we have found that scientific video, combining animated images with sound, is of great help. We will describe some of the techniques we use to generate images and sound, trying to emphasize those that have not yet become common.

Although graphics hardware has become increasingly powerful, renderings of complex scenes with proper shading and lighting are still difficult to generate in real time for moderate cost. As a result, we employ interactive techniques where they are essential for data analysis, such as scatterplot brushing [2] and selecting a viewing perspective or position of a light source. Our model has been to generate high-quality rendered images that are then recorded a frame-at-a-time on NTSC videotape; by high quality, we mean anti-aliased perspective images including texture maps, lighting models, mild reflectivity, and transparency. Once an animated sequence has been generated on videotape, we then construct a synchronized "sound track".

We have presented some of our basic image tools elsewhere [5]. Our most important tool is the equi-spaced color-level plot with either orthographic or perspective projections. Here, we will describe a dashed surface plot that can simultaneously convey the shape of a two-dimensional function and its local error.

We also presented our basic sound tools in [5]. We have used sound in several forms. Sound has been most useful to underscore the passage of time in the form of beats. We have also found it useful to vary the pitch, volume, and tempo in order to represent other scalar quantities. Here, we will describe means based largely on variation-diminishing splines for stretching, smoothing, or compressing data to fit into a prescribed video segment. In addition, we explain how "loudness equilibration" can help make listener perception more uniform.

Our view is that both the monolithic system and the subroutine model of software communication are inappropriate for this application. It is often the case that data must be transmitted between machines or highly specialized programs. Hence, it is attractive to employ standardized, self-descriptive, ASCII file formats and use files or UNIX pipes to transmit data between disjoint processes. All of our tools use a uniform interface for exchanging data [4], based on the AWK paradigm [1]. We employ the RenderMan Interface Bytestream [12] (RIB) to decouple the modeling and rendering tasks while preserving reasonable generality in possible graphical techniques.

The next section (§ 2) presents our schemes for generating images. In § 3, our techniques for generating and manipulating sounds are discussed.

## 2 Colored and transparent surfaces for fields

The basic tool for understanding two-dimensional images is the color-level plot. This is the natural generalization of line-based contour plots adapted to modern raster graphics hardware that supports texture mapping. Orthographic projection results in a color-level plots that are close to traditional contour plots (using gray-scale instead of color brings one even closer). The eye does not respond uniformly in wavelength to light. Hence, it is important to choose colors that appear to be equi-spaced. This process can be reduced to a nonlinear least squares in an appropriate psychophysical metric [5].

A flat orthographic color-level plot is not the most intuitive representation for a two-dimensional data field, though in trained hands it is often the most informative. We have found that a perspective projection helps a great deal, particularly when the surface is given specular highlights. The choice of perspective is often best made with an interactive tool; we have described a "helicopter" model elsewhere [5]. Shading provides cues about inflections and other subtle phenomena.

Often a sequence of two-dimensional data fields is provided. The successive images may represent data at different times or as a function of another parameter. We employ frame-at-a-time animation to generate a video segment representing the data, where each frame is typically a color-level plot.

Our target medium is VHS videotape since, for now, that is the only universally presentable format. We have attended many meetings where the speaker complained that some critical feature was impossible to see in the displayed video but was quite clear in his lab. So we try to use all available techniques (such as anti-aliasing, color desaturation, and motion) to make legible at least the most important features. Forcing ourselves to stay with NTSC resolutions has also helped us avoid the tendency to add distracting dials and other dynamic icons to already complicated images.

*Loess* is a general mechanism for computing a smoothed approximation to scattered data, which produces local standard error estimates [3]. One new approach to viewing a *loess* surface and the local errors is the *dashed surface*. In two dimensions, the *dashed surface* is constructed by dividing the domain into squares and then trimming the color-level plot in each square in proportion to the error. That is, look at the a local square patch of the surface. The colored region of the square is retracted from the edges as the error increases. This trimming of the plot in the square must be done so that the patch area (not the perimeter) is inversely proportional to the error. This approach results in a plot that "breaks up" where the smoothed representation is not good; it is the natural generalization of a one-dimensional plot that becomes broken up into smaller and small dashes as the error increases. We reserve the use of error bars projected up from the surface for displaying actual residuals. See Figure 1.

## 3 Sound for scalar parameters

The use of sound is still rather speculative. In fact, many visually oriented people may question why it should be preferred over a more complex image. We would like to argue the case for sound to represent scalar parameters associated with an animated sequence of data fields. Examples of such scalar parameters are time, global error values, and functions computed from the data fields.

The simplest and most effective application of sound is to denote the passage of time in a simulation. Drumbeats are the natural sound to associate with a discrete moment in time and are analogous to the tick marks found on a scatterplot. (The same software that we use in our graphics software to pick "round" numbers for tick marks was immediately applicable to generating such time beats.) We found earlier that beats attached to a particular simulation hesitated at points where time steps had to be repeated. We had overlooked the repetitions of time in a laserprinter plot of the time progression; the beats stretched out the time points making it possible to hear things we had not seen earlier.

We have also considered the use of pitch, volume, and tempo to represent more complicated scalar functions or a combination of functions [5]. In one example, we had a scalar function of another scalar parameter. We varied the pitch to represent the "independent" scalar while increasing the volume and repetition rate in proportion to the "dependent" variable.

Our experience with auditory representations of more complex scalars is that they cannot replace line drawings. We have seen a number of animations where time (or another scalar) is represented by an analog clock or a number displayed on the periphery of the basic data field display, such as our color-level plots. (Variable length bars are a more workable alternative if there are only one or two bars.) Introducing extraneous visual information is often distracting — the eye must glance from the main display to the indicator and important information may be missed. If you examine a line drawing of the scalar parameters carefully, then the sound representation gives a qualitative impression of the scalar value without the annoying distraction. The other problem with adding scalar values as clocks or bars in the image is the limitations of

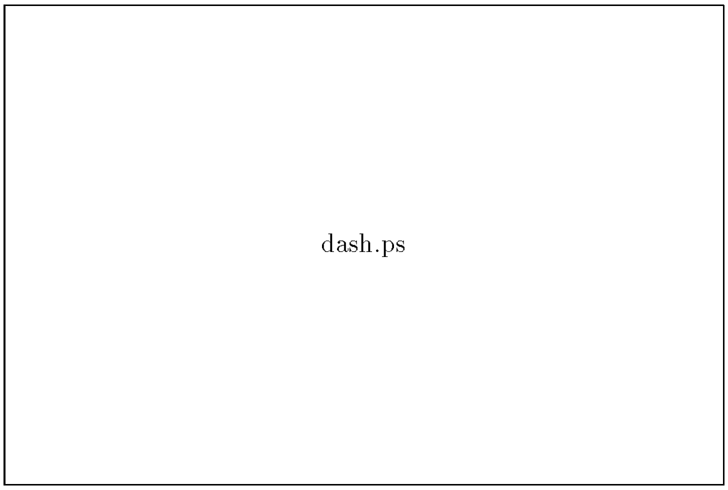


Figure 1: Qualitative display of standard errors by using a dashed surface.

NTSC resolution; we have found that the basic images are usually complex enough without trying to add other small auxiliary features. (Obviously, this latter comment doesn't apply to high resolution displays but the distraction comment does.)

Sound suffers from some of the same limitations as color: it has limited resolution, is not as familiar as conventional line drawings, and has some hard pyschophysical problems. In addition, sound is only applicable to animated sequences since it's inherently transient. Finally, human beings can assimilate much more information visually than via hearing so sound must be employed in limited ways.

Let us consider some of the parameters governing auditory perceptions and how we use them:

- pitch: If we constrain ourselves to the Western scale, we can take half-steps over three to five octaves. Although the use of major scales is more pleasing to the ear, it severely reduces the number of available notes and, hence, the resolution. Many (synthesized) instruments are incapable of generating notes over a wide frequency domain. We have considered trills to represent error bars but so far have found the technique to be of limited value.

- tempo: By this, we mean the duration and frequency of striking notes. Humans are surprisingly sensitive to variations in tempo.

- volume: It is possible to vary the volume over a number of relatively fine steps but perceptions are coarse. This problem is made worse by the fact that a particular fixed amplitude will be heard to have a different "loudness" as the pitch is varied.

- voice: We use this term to mean instrument. It is possible to distinguish and follow the notes generated by several instruments. We sometimes supplement this using stereo and reverberation to emphasize the separation of voices.

- melody: We have considered transitional notes and chording patterns to denote changes in scalar values, but have not been completely satisfied with the results at present. See [10] for a discussion of melody versus chord.

This variety of knobs would in principle allow the simultaneous presentation of many scalar variables. We have had better luck by instead presenting only one or two variables and using them to control several musical parameters; the redundancy overcomes some of the psychophysical difficulties.

We have built a number of tools to generate sounds based on our tensor/scatter file format [4]. The tools can generate percussion to denote time, percussion to act as a counter of discrete events (heard in the associated videotape to count number of Newton iterations), and a variety of more complex sounds with variable pitch, volume, and tempo. All of the tools generate standard MIDI [8] files to drive our synthesizer equipment [9]; MIDI allows the specification of events in time that start (or stop) a note with a particular velocity (roughly volume).

We have found that our generic approximation tools also play a role in sound generation. Often we will have a fixed sequence of scalar values that are uniformly spaced. Such a sequence may have to be translated into more than one similar sound sequence lasting different lengths of real time. Variation-diminishing splines can be used to expand, compress, or resample the sequence in a manequiloud.d.ps

Figure 2: Plot of perceived dB level (60 phons) as frequency is varied.

ner suited to the sound segments of different lengths; for data with dramatic excursions (outliers), such variation-diminishing techniques also smooth the data so the sound is more palatable and, more importantly, easier to grasp (given the limitations of the ear). For non-uniform data, least-squares spline fits or *loess* can be employed.

We have experienced difficulties with loudness perception. As mentioned earlier, the human ear perceives notes of varying pitch but equal amplitude as having different loudness values. Figure 2 is based on measured data [11, p.45] and clearly shows that low and high frequency response is not flat. MIDI defines middle-C as note 60 so we can map note $n$ onto frequency by

$$f = 2^{(n-60)/12} 523.25.$$

We can then build a loudness compensation function by using the above formula and logarithmic interpolation of the measured data from fig. 2. We have done this and find that it does improve volume perception — this permits us to use a wider range of frequencies and, hence, increase the resolution available with sound. The technique should be extended to include a second variable varying the amplitude (other phon values). Other possibilities would be to incorporate alternative loudness measures like $L = kI^{0.3}$ where $L$ is loudness and $I$ is intensity and to include temporal integration [11].

Another difficulty is that MIDI only lets us directly control the "velocity" and that is only loosely related to loudness for some instruments. Among the voices we use, piano and the xylophone seem about the most linear.

Sound is limited to providing a small number of cues for scalar parameters. We have found it possible to mark time with drumbeats and then use different instruments with variable pitch and so forth to track two or three scalars. Using more voices for more parameters seems ineffective.

There is a growing literature on the subjects discussed here. For example, other related papers in the proceedings that contain [5] are [6] and [7].

## Acknowledgements

We thank Mark Kahrs and Tom Killian for supplying music generation and synchronization tools. RenderMan is a registered trademark of Pixar. Unix is a registered trademark of Unix Systems Laboratory, Inc.

# References

[1] Alfred V. Aho, Brian W. Kernighan, and Peter J. Weinberger. *The AWK Programming Language.* Addison-Wesley, New York, 1988.

[2] Richard A. Becker and William S. Cleveland. Brushing scatterplots. *Technometrics*, 29:127–142, 1987.

[3] William S. Cleveland and Eric Grosse. Computational methods for local regression. *Statistics and Computing*, 1:1, 1991.

[4] W. M. Coughran, Jr. and E. H. Grosse. A philosophy for scientific computing tools. *SIGNUM Newsletter*, 24:2–7, 1989.

[5] W. M. Coughran, Jr. and E. H. Grosse. Techniques for scientific animation. In E. Farrell, editor, *Proceedings of the 1990 SPIE/SPSE Conference #1259: Extracting Meaning from Complex Data*, pages 72–79. The Society for Imaging Science and Technology, 1990. Associated videotape in 1259-V collection.

[6] S. P. Frysinger. Applied research in auditory data representation. In E. Farrell, editor, *Proceedings of the 1990 SPIE/SPSE Conference #1259: Extracting Meaning from Complex Data*, pages 130–139. The Society for Imaging Science and Technology, 1990.

[7] G. G. Grinstein and S. Smith. Perceptualization of scientific data. In E. Farrell, editor, *Proceedings of the 1990 SPIE/SPSE Conference #1259: Extracting Meaning from Complex Data*, pages 190–199. The Society for Imaging Science and Technology, 1990.

[8] International MIDI User's Group. MIDI specification. Technical report, P.O.Box 593, Los Altos CA 94022, 1983.

[9] T. J. Killian. Computer music under the 10th Edition UNIX System. In *UNIX Research System Papers, Tenth Edition Volume II*, pages 477–482. Saunders College Publishing, 1990.

[10] David Lunney and Robert C. Morrison. Auditory presentation of experimental data. In E. Farrell, editor, *Proceedings of the 1990 SPIE/SPSE Conference #1259: Extracting Meaning from Complex Data*, pages 140–146. The Society for Imaging Science and Technology, 1990.

[11] B. C. J. Moore. *An Introduction to the Psychology of Hearing.* Academic Press, London, 2nd edition, 1987.

[12] Pixar. *The RenderMan Interface, Version 3.1.* 1989. 3240 Kerner Blvd, San Rafael CA 94901.