Display of grey-scale data on a bilevel output device

Steve Cole and Joe Dellinger

ABSTRACT

Graphics display devices that allow the intensity of each pixel to be varied are ideal for displaying seismic data, since the intensity of each pixel can indicate the amplitude of the corresponding sample of the seismic data set. Hardcopy devices do not allow the intensity of individual pixels to be varied — each must be "on" or "off". It is possible to achieve the effect of intensity variation on such devices by controlling the spatial density of "on" and "off" pixels. Each pixel in the continuous-tone image is compared to a threshold value. If the input intensity is greater than the threshold, the corresponding pixel of the output device is set to "on". Otherwise it is set to "off". The methods available for accomplishing this differ in the way that threshold values are obtained. We find these techniques capable of generating hardcopy images that are similar in quality to graphics screen images. For many applications, these displays are preferable to conventional "wiggle trace" displays of seismic data, which overcome the lack of intensity variation by using the amplitude of the wiggle trace to represent the amplitude of the corresponding sample of the seismic data set.

INTRODUCTION

High-resolution graphics display devices are useful tools for looking at seismic data. Such devices have a large number of pixels and the ability to vary the intensity of each pixel over a wide range, either in color or on a grey scale. By using the intensity of a pixel to represent the amplitude of a sample from a seismic data set, such a device can readily display two or three dimensional seismic data. At SEP, we are fortunate to have had for some time excellent software for displaying movies of such data sets (Ottolini *et al.*, 1984), but the ability to produce a copy of such movies on paper was limited by the fact that hardcopy devices cannot vary the intensity of pixels in the same way as display devices.

Graphics displays typically use eight or more bits of memory to define the color of each display pixel. On a monochrome display with eight bits per pixel, such as the Rastertek display used at SEP, each pixel can have any of 256 different grey-scale intensities, ranging from 0 (black) to 255 (white). We consider the problem of converting an image displayed on such a device to bilevel form, where each pixel is either black or white, suitable for plotting.

As a first step, consider this conversion problem for the case where there is a one-to-one correspondence between pixels on the display and plotter. For each pixel, we must represent a grey-scale value ranging from 0 to 255 by a 0 (black) or 1 (white). The best that can be done is to choose a threshold value between 0 and 255, then set the output pixel to 1 (white) if the intensity of the input pixel is greater than the threshold value, or set the output to 0 (black) otherwise. Selecting and applying a threshold is common to all methods used to perform the conversion. What distinguishes the methods and determines how well they perform is the way thresholds are chosen.

If there is a one-to-one correspondence between input and output pixels, seven-eighths of the information is lost in the conversion, and image quality is bound to suffer. To enhance the quality of the output image, each input pixel can be mapped to a number of points on the output device. For example, suppose that each input pixel is mapped to a four by four rectangle of pixels on output. Having sixteen output pixels for each input will yield an output image with more bits of information than the input. The following discussion will show that simply conserving or increasing the amount of information is not enough to insure a good output image. Thresholds must be chosen carefully to take advantage of the extra bits.

COMPARISON OF PROCESSING METHODS

This section describes some methods commonly used to select threshold values for converting a grey-scale image to bilevel form. To assess the effectiveness of different methods, we have constructed a "test pattern" containing a portion of a stacked section, a photograph, a grey scale, and a sinusoidal pattern. These images help illustrate how the various processing schemes affect different types of data.

Fixed threshold

In this simple scheme, a single value is used as a threshold for the entire data set. Constant-intensity regions come out all black or all white depending on whether their intensity is greater than or less than the threshold. The method therefore yields only black and white regions, with no apparent grey areas. This method fails to take advantage of the mapping of one input pixel to a number of output pixels because the threshold levels for all the output pixels are the same. Figure 1 shows the test pattern processed with a fixed threshold of 128. Despite the poor grey-scale resolution, the major features of the seismic data and photograph are still easy to see. The white and black bar beneath those two images is actually a linear grey scale, handled poorly by the primitive processing. The shape of the sinusoidal patterns beneath the grey scale is handled well by this method because of the lack of significant grey information.

Random dither

Dithering is a term used to describe methods where the threshold value is varied. A simple dithering method is random dither, where the threshold value is randomly selected from the range 0-255 at each point. The result of applying random dither to the test pattern

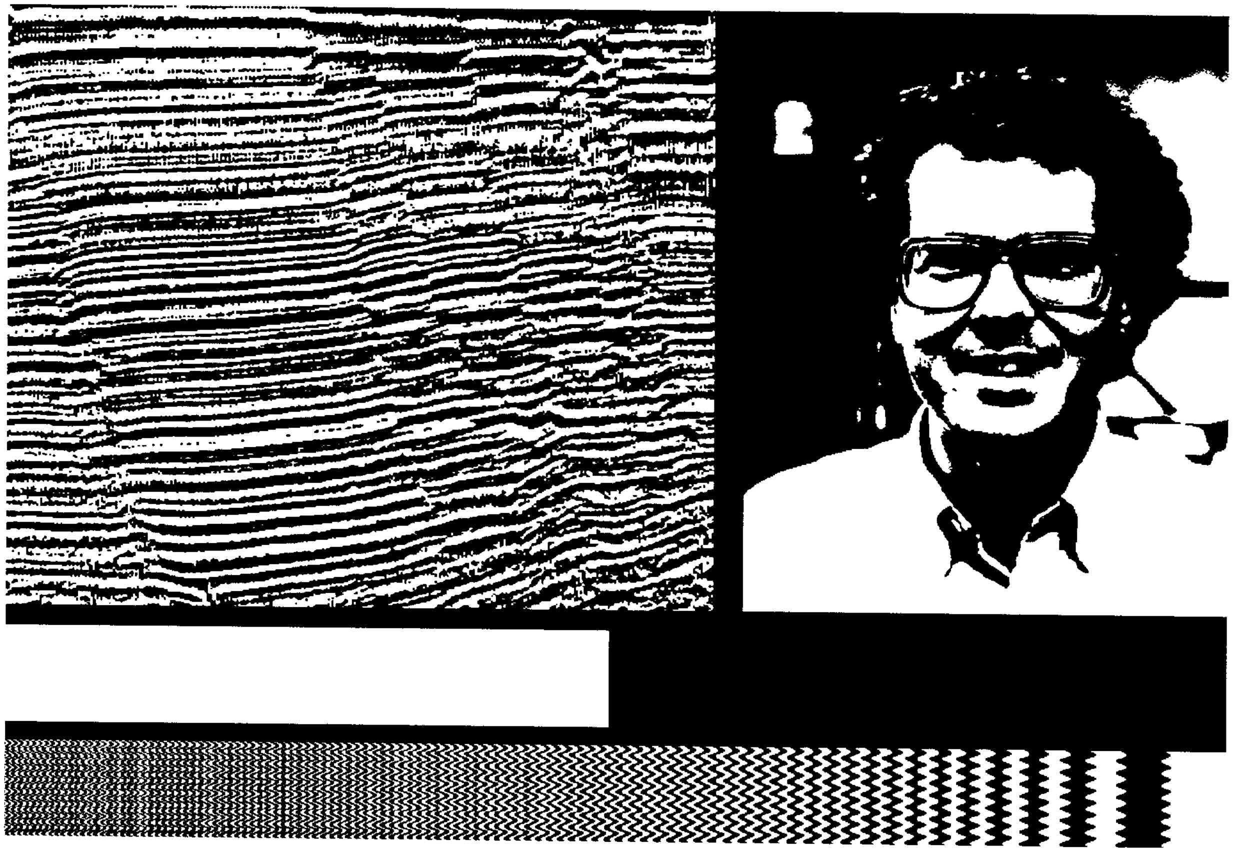


FIG. 1. Test pattern converted to bilevel format using a fixed threshold of 128.

is shown in Figure 2. The varying threshold provides better grey-level resolution — many shades of grey are now visible. However the random thresholds cause constant-intensity regions to appear non-uniform, resulting in a noisy image. In the past, this method has been used at SEP for generating hardcopies of seismic movies (Sword *et al.*, 1986).

Ordered dither

In ordered dither, the threshold is varied in a systematic way. A square dithering matrix is superposed on the data in checkerboard fashion, so that the threshold used to determine whether any point is "on" depends simply on the position of the point within the data set. Consider again the case where each input pixel is mapped to a four by four array of pixels on output. If a four by four dithering matrix (containing sixteen different thresholds) is used, then for each input pixel the output is a four by four rectangle that can have any of seventeen different apparent grey-level intensities, ranging from the case where 0 pixels are set "on" to the case where all sixteen are on. Note that if the intensity of the input pixels ranges from 0 to 255, the sixteen thresholds should also cover this range and should be equally spaced to provide the most accurate grey-scale reproduction.

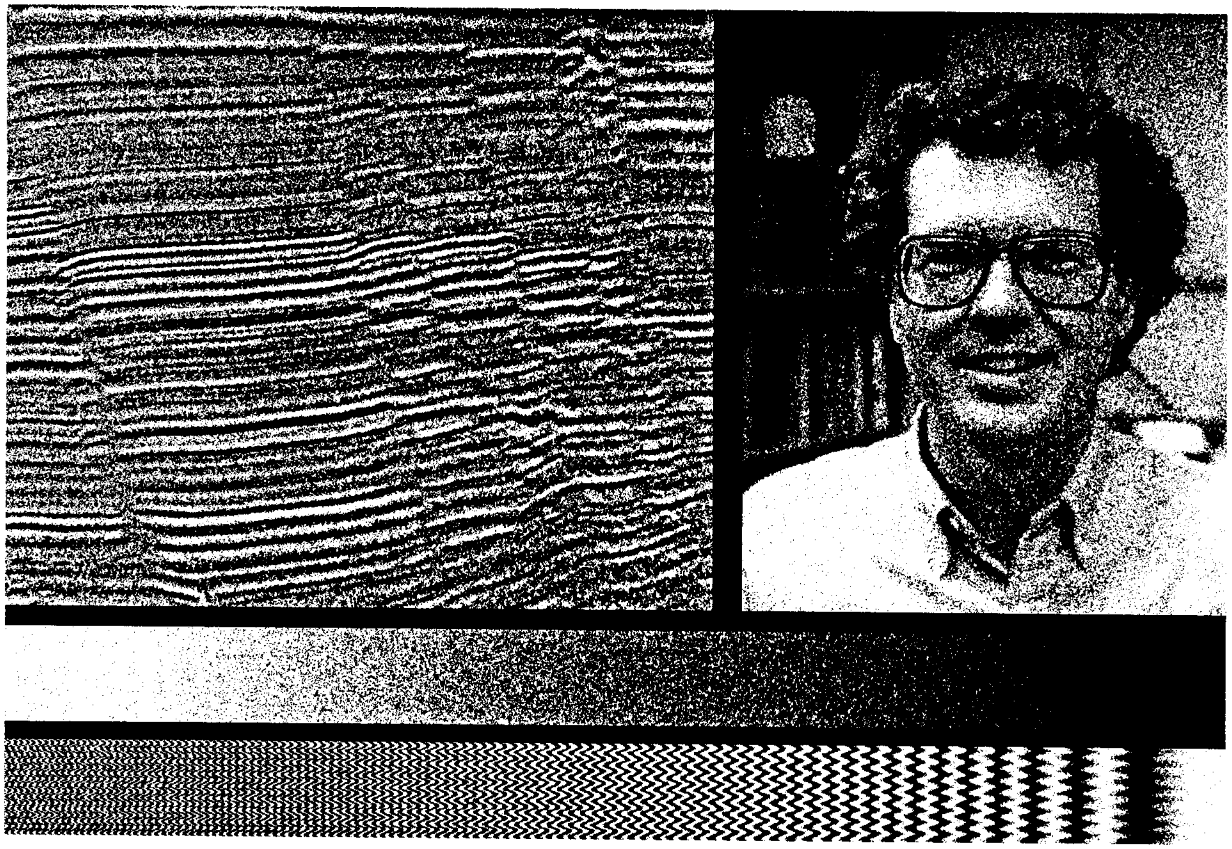
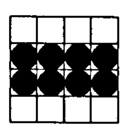


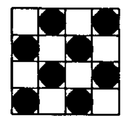
FIG. 2. Test pattern converted to bilevel format using random dither.

It is readily apparent that the ordering of the threshold levels in the matrix is of great importance when designing dithering matrices. Consider a four by four dithering matrix. If thresholds are arranged such that the eight smallest thresholds occupy the middle two rows of the matrix, then a constant input level of 128 would yield on output the pixel matrix:



A constant-intensity region with a grey-scale intensity of 128 would be filled with distracting horizontal stripes.

To prevent the introduction of such objectionable patterns in the conversion process, similar threshold values must be far apart. For example, the dithering matrix used above may be modified so that the eight smallest thresholds are separated by at least one other element in the horizontal and vertical directions. Then a constant input intensity of 128 would yield on output the pixel matrix:



This result, while regular, has no lines or other clusters of pixels that will interfere with the image. It produces a diffuse grey pattern.

Note that when comparing neighboring thresholds, it is necessary to consider wraparound effects, since the dithering matrix is applied to the data in a checkerboard pattern.

To further illustrate the need for careful ordering of the thresholds in a dithering matrix, Figure 3 shows the result of applying a randomly-ordered sixteen by sixteen dithering matrix to the test pattern. Note the pattern superimposed on the data. This pattern is caused by the poor positioning of dithering matrix elements. If two similar thresholds are placed together, then in most cases output pixels corresponding to those thresholds will have the same polarity - both black or both white.

Figure 4 shows the test pattern processed with a sixteen by sixteen dithering matrix where similar thresholds are separated as widely as possible. This separation eliminates the pattern of clusters of like points that was noticeable in Figure 3.

An algorithm for generating optimal dithering matrices such as the one used to produce Figure 4 was developed by Jarvis, Judice, and Ninke (1976). It is based on the criterion that similarly-valued thresholds should be as far apart as possible in the matrix. The algorithm begins with the two by two dithering matrix that obviously satisfies the criterion:

$$D^2 = \begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix}$$

and gives a *recurrence relation* for generating an n by n dithering matrix, where n is a power of 2:

$$D^n = \begin{pmatrix} 4D^{n/2} + D^2_{00}U^{n/2} & 4D^{n/2} + D^2_{01}U^{n/2} \\ 4D^{n/2} + D^2_{10}U^{n/2} & 4D^{n/2} + D^2_{11}U^{n/2} \end{pmatrix}$$

where $U^{n/2}$ is the n/2 by n/2 matrix containing all ones. Applying the recurrence relation to D^2 gives the four by four dithering matrix:

$$\begin{pmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{pmatrix}$$

Note that the matrix elements shown here should be scaled to cover the range 0-255.

It is not necessary to map each input pixel to a fixed number of output pixels. Such a limitation would restrict the technique to producing images of only certain fixed sizes, and is not a limitation imposed by our implementation of ordered dither. In fact, our

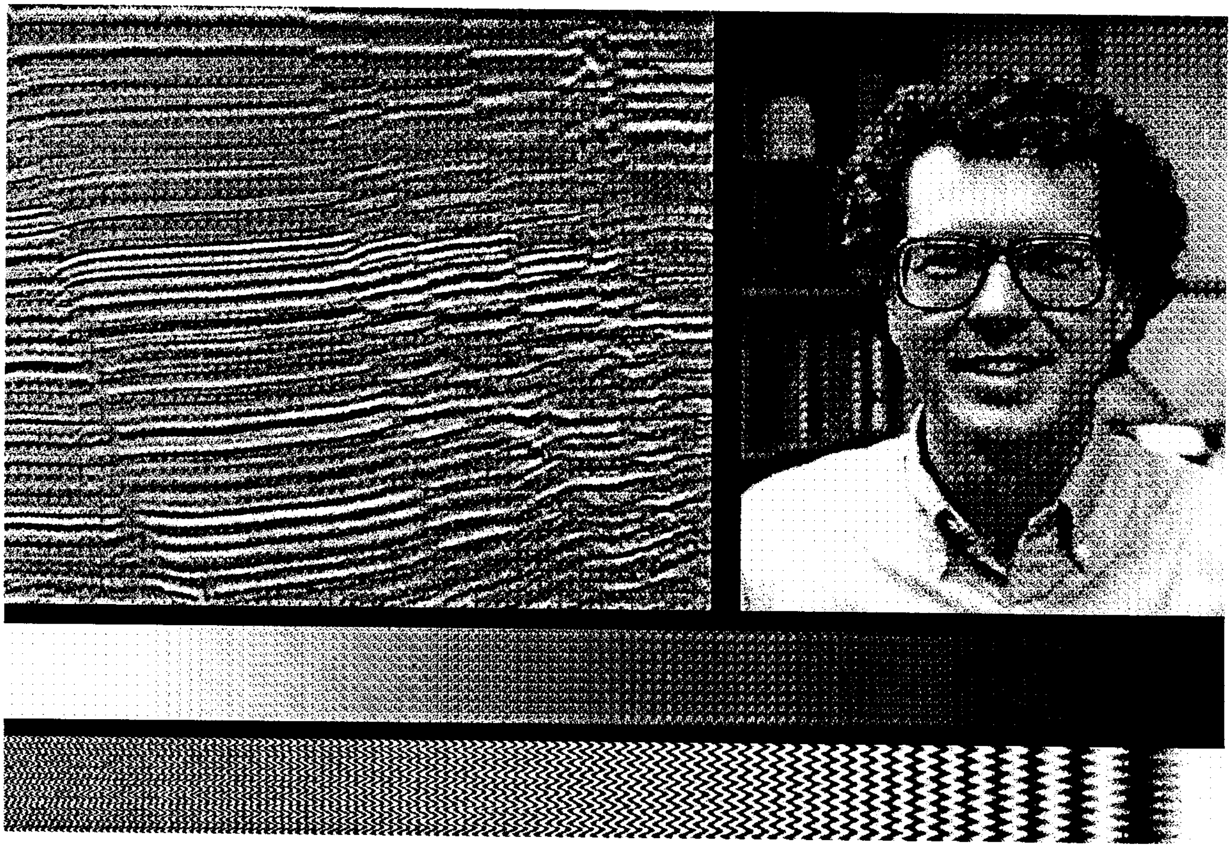


FIG. 3. Test pattern converted to bilevel format using a randomly-ordered 16 x 16 dithering matrix.

implementation is extremely general. For a given input image, there are no restrictions on the size of the output image. Our software stretches the input image to the desired size, then applies the dithering matrix. Typically we use a sixteen by sixteen dithering matrix. Such a matrix provides the best possible resolution of grey levels since each threshold 0-255 appears in the matrix exactly once.

To illustrate the flexibility of our implementation of ordered dither, the test pattern, as displayed on our Rastertek, contains 985 pixels in the horizontal direction and 646 pixels in the vertical direction. The figures included in this report are 6.5 inches by 4.25 inches. Our Imagen laser printer plots 300 dots per inch, thus the image displayed in the report is 1950 pixels by 1275 pixels. For each input pixel, then, the output image contains an average of approximately 3.9 pixels. The output image contains only about half the information of the input image. In our opinion, though, it retains all the important features of the image as displayed on the graphics screen.

FIG. 4. Test pattern converted to bilevel format using an "optimal" 16 x 16 dithering matrix.

Minimized average error method

A feature common to the algorithms presented thus far is that they are data-independent. Thresholds are determined randomly or by position in the image, but not on the intensities present in the image. The possible error using any of these methods is large. For example, suppose the input image happens to have an intensity at each point that is one smaller than the corresponding dithering matrix element. All the pixels in the output image will be set to "off", despite the fact that the image actually contained intensities as large as 254! Clearly there are cases where an algorithm that takes the input intensities into account could yield a better representation of the data.

A simple method that does adapt to the data is the *minimized average error method*, developed by Floyd and Steinberg (1975), and nicely described by Knuth (1987). The idea here is that the error between the input and output images at a particular point is taken into account when converting neighboring points to bilevel form.

A *Ratfor* description of the algorithm helps to further describe the method:

```
#Floyd-Steinberg algorithm
integer grey(n1,n2) bilev(n1,n2)
integer n1,n2,i1,i2
real alpha, beta, gamma, delta, error
alpha = 0.4375; beta = 0.1875; gamma = 0.3125; delta = 0.0625;
do i1=1,n1 {
   do i2=1,n2 {
      if (grey(i1,i2) < 128) then
         bilev(i1,i2) = 0
      else
         bilev(i1,i2) = 1
      error = grey(i1,i2) - 255 * bilev(i1,i2)
      grey(i1,i2+1) = grey(i1,i2+1) + error * alpha
      grey(i1+1,i2-1) = grey(i1+1,i2-1) + error * beta
      grey(i1+1,i2) = grey(i1+1,i2) + error * gamma
      grey(i1+1,i2+1) = grey(i1+1,i2+1) + error * delta
      }
   }
stop
end
```

Figure 5 shows the result of using this algorithm to convert the test pattern to bilevel form. The improvement over the ordered dither result of Figure 4 is significant. In general, the Floyd-Steinberg method seems to give much sharper images than ordered dither. The faint error diffusion patterns visible in constant-intensity areas of the test pattern are less objectionable than the patterns imposed by ordered dither. In many areas, the error diffusion patterns seem to "flow" from the upper left to the lower right. We believe that this is caused by the direction of error propagation in the algorithm. Varying the direction of error diffusion would be complicated from a programming standpoint, but might make the error diffusion patterns less noticeable.

Because it is a more sophisticated algorithm than ordered dither, it is not surprising that the Floyd-Steinberg method is more expensive. However, the difference in cost is not prohibitively large. On our Convex C-1 computer, ordered dither required approximately one minute of CPU time to convert the grey-scale test pattern to bilevel form, while the Floyd-Steinberg method required about one and a half minutes.

ENHANCEMENTS

If the conversion to bilevel form is performed according to the discussion of the previous section, the bilevel images do not compare well to the original grey-level images. When viewed on the Rastertek display, the bilevel images seem to have much poorer contrast than the original grey images. When plotted on our Imagen laser printer, the bilevel images seem much too dark. The explanations for and solutions to these two problems reveal some interesting facts about graphics hardware and vision.

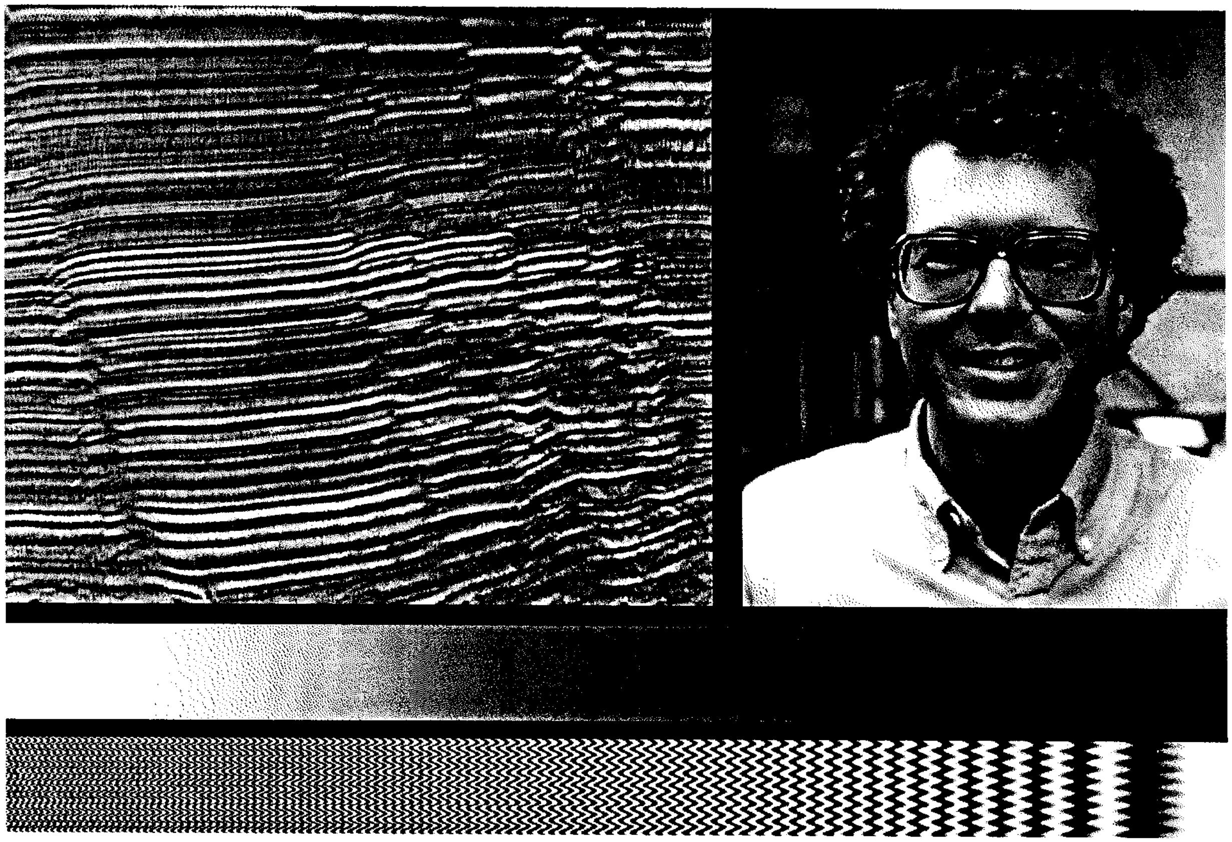


FIG. 5. Test pattern converted to bilevel form using the Floyd-Steinberg or minimized average error method.

Pixel overlap correction

The conversion methods discussed earlier have an underlying assumption that images, either on a graphics screen or on paper, are composed of an array of pixels that fit together nicely, with no empty space or overlap. This assumption is not valid, and it is easy to show that techniques such as ordered dither need to be modified.

To illustrate this point, consider an area of a graphics screen that has all its pixels set to an intensity value of 128, the center of the grey scale. Ordered dither dictates that to recreate this grey area on a plotter, simply turn on every other pixel in a checkerboard fashion. If all the pixels fit together with no empty space or overlap, then 50% of the paper will be covered by black ink, giving an appearance identical to the screen where each pixel is set to half its full-scale intensity.

In the left half of Figure 6, the Floyd-Steinberg algorithm was applied to a series of constant-intensity stripes. The intensity increment between stripes is constant, so the transition from black to white should appear linear. Yet clearly it is not. Several of the dark stripes seem to be roughly equal in intensity, then the transition to white occurs rapidly.

For the moment, consider the right half of Figure 6 to be a representation of the same horizontal stripes as they appear on the graphics screen. Note the desired gradual change in intensity from one level to the next. The correction that enables us to reproduce this figure on paper will be discussed shortly.

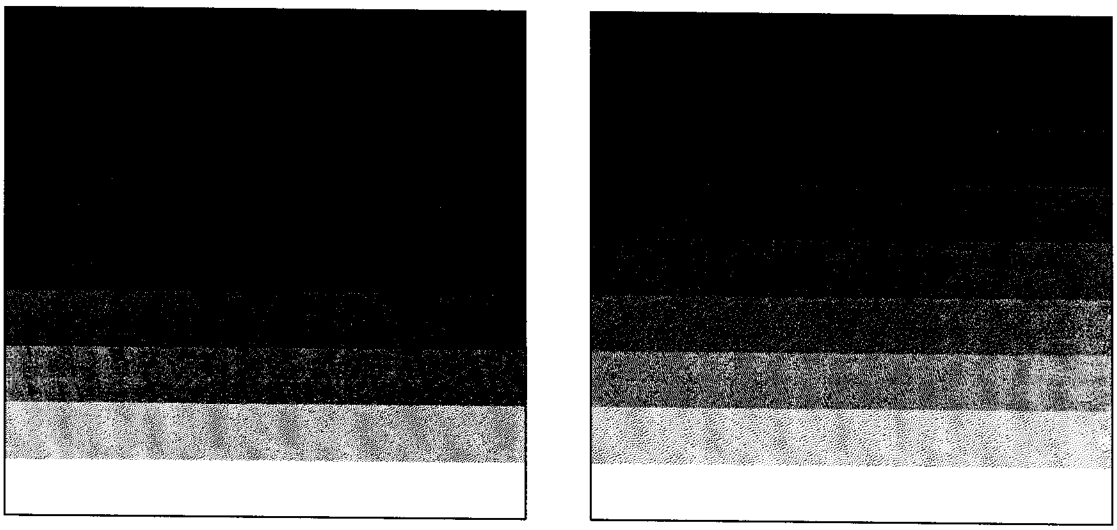
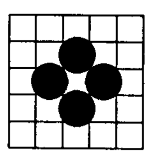


FIG. 6. A series of constant-intensity bands, with an equal intensity increment between bands, converted to bilevel form by the Floyd-Steinberg algorithm. On the left, note a perceived non-linearity, particularly for the dark bands. On the right, the pixel overlap correction discussed in the text has been applied to restore the linear appearance.

Since it was clear from early tests that a linear intensity variation on the graphics screen did not always translate to a linear variation on paper, it seemed natural to question our assumption that a plotter produces a rectangular array of pixels with no leftover space or overlap. As a first guess, we hypothesized that the individual dots produced on paper were round instead of square, and because of their rounded shape they exceeded their square area on the page as in the following figure:



Note the white square surrounded by dots on all four sides. The overlap of the dots due to their round shape has caused much of this square, which should be white, to be covered by ink. Returning to the example of ordered dither applied to a region having a constant grey-scale intensity of 128, note that all the empty squares, being surrounded by four black dots, will have this problem. Viewed on a large scale, the area will be much darker than it should be.

It is easy to predict the magnitude of this darkening as a function of intensity and to scale the data to correct for it before converting to bilevel form. For ordered dither, Figure 7 shows the perceived intensity, or amount of white area , as a function of the input grey-scale intensity. As the input intensity decreases from 255 (white), no overlap occurs until the midpoint of the intensity scale is reached. At that point, half the pixels have been set to black in the familiar checkerboard pattern. When any additional dot is placed on the page, it will overlap with its four neighbors, and the intensity change will be smaller than expected due to the overlap.

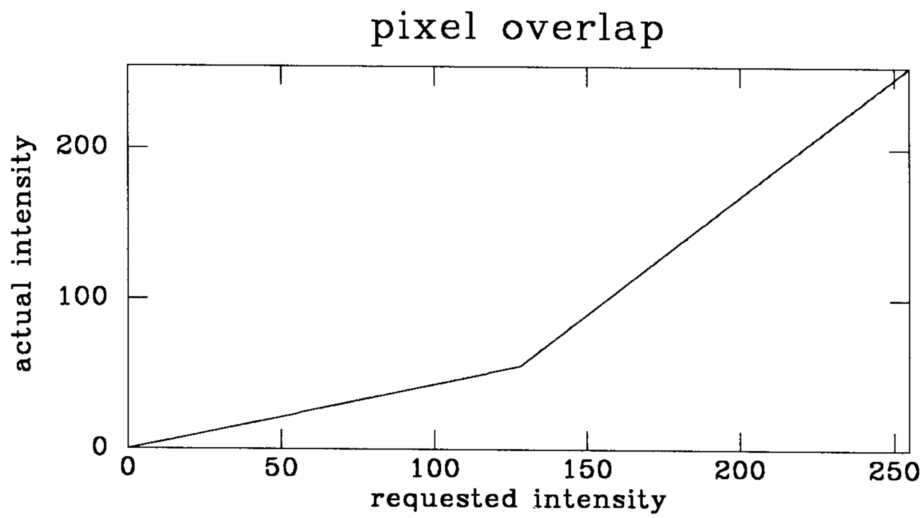


FIG. 7. Plot of perceived intensity (amount of white area on paper) as a function of requested intensity (number of white pixels) for ordered dither. 0 is black and 255 is white. Below 128, new black dots are overlapping with all four neighbors, so the intensity change is more gradual.

Dithering can be made to take the overlapping into account. The input amplitudes are simply scaled such that, given the intensity change due to overlapping shown in Figure 7, the perceived intensity (or area of the page covered with ink) will be correct. This correction was used to produce the right half of Figure 6. Since the correction produces an image that is quite similar to the graphics screen image, we feel that our hypothesis of pixel overlap is correct.

Note that the pixel overlap correction as described here was applied in Figures 1 through 5. Figure 8 shows the test pattern processed by the Floyd-Steinberg algorithm, but with no overlap correction. Compare to Figure 5.

Nonlinearity of graphics displays

The pixel overlap correction made the plotted image more similar to that displayed on the graphics screen, yet there was still an obvious difference between the two. The plotted image seemed to have less contrast (fewer pixels at the two ends of the intensity scale). This "washed out" appearance made it more difficult to identify features on the plotted image. We believe that there are two reasons for this difference, and have empirically arrived at a correction that, for our combination of graphics display and plotter, allows us to reproduce screen images extremely well on paper.

The first reason involves the human visual system. It is well known that our eyes do not

FIG. 8. Test pattern converted to bilevel form using the Floyd-Steinberg algorithm. The pixel overlap correction was not applied to this figure, thus it is much darker than Figure 5.

respond linearly to changes in light intensity. In fact, it has been observed (Pratt, 1978) that a linear increase in light intensity yields a logarithmic increase in perceived brightness. Thus on a graphics display a linear grey scale will appear to change from black to grey rather quickly, then slowly change from grey to white.

The second reason for a perceived nonlinearity in graphics screen images is the device itself. The efficiency with which the display will represent a requested intensity level varies from device to device. We have observed differences in the display of a linear grey scale between our Rastertek graphics display and Sun workstations, and suspect that this device-dependent efficiency is the cause. It is difficult to quantify because we cannot separate this effect from the nonlinearity caused by our eyes.

While we are only guessing that these effects are the underlying reasons for the perceived nonlinearity of linear intensity changes on our graphics displays, it is simple enough to design a correction that simulates this nonlinearity on paper. We find it desirable to simulate this nonlinearity because the additional contrast obtained by having more energy at the ends of the intensity range makes it easier to distinguish features in the images.

To design such a correction, we noted that, after applying the pixel overlap correction discussed earlier, the reproduction of the middle of the intensity range (grey levels with an intensity around 128) appeared correct, and that what was needed was simply to apply a gain on either side of 128 to direct more of the pixels toward the extreme values of 0 and 255. We designed a cubic polynomial that leaves intensity values 0, 128, and 255 unchanged, and adjusted the one remaining free parameter until the results looked correct, which determined the correction displayed in Figure 9.

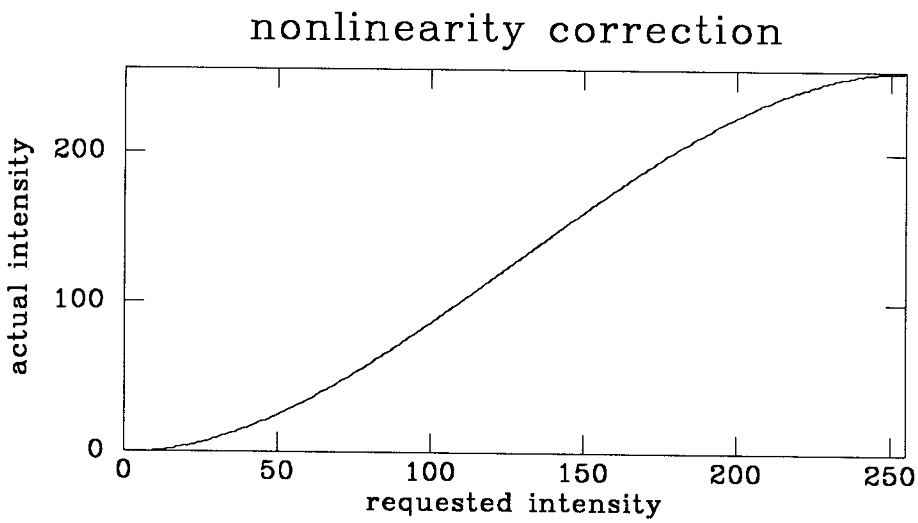


FIG. 9. Correction to simulate on paper the perceived nonlinearity of images displayed on our best graphics device. The correction is machine dependent; this particular function yields results on our Imagen laser printer that look like those on our Rastertek display.

To illustrate the importance of this correction, Figure 10 shows the result of converting the test pattern to bilevel form without applying this nonlinearity correction. This figure has very little information close to the ends of the intensity range (pure black or white), as compared to Figure 5, where the nonlinearity correction was applied. We think that in addition to giving closer agreement with our graphics displays, the nonlinearity makes it easier to see details in Figure 5 than in Figure 10. Note that this correction was applied in Figures 1 through 4 as well.

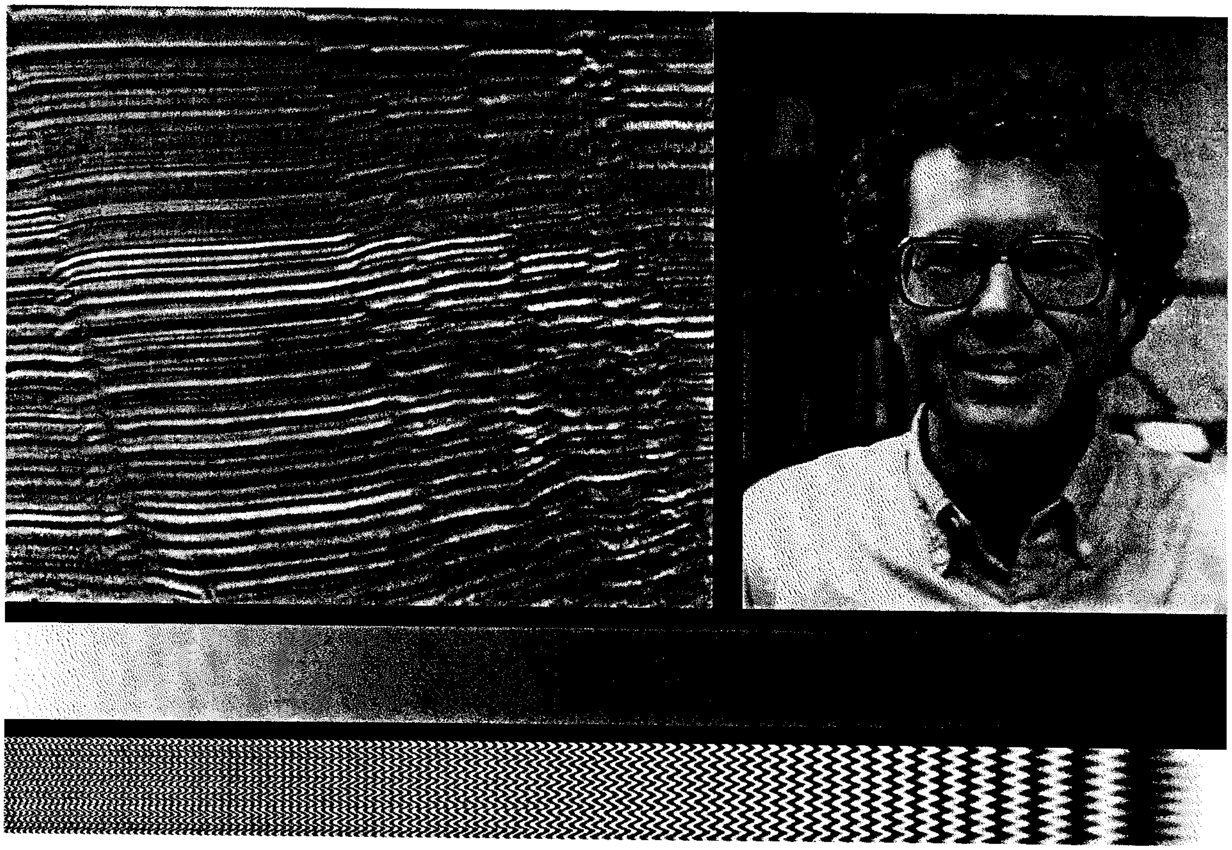


FIG. 10. Test pattern converted to bilevel form by the Floyd-Steinberg method. The correction to simulate the nonlinearity of display devices was not applied. Compare to Figure 5, where the correction was applied.

APPLICATIONS

We believe that the techniques described here offer some important advantages when applied to plotting seismic data. First, they provide a high-quality way to reproduce data displayed on a graphics screen. Moreover, we think that this method offers some advantages over the traditional "wiggle trace" display.

Figure 11 shows a shot gather plotted using the wiggle trace display and the techniques developed in this paper, which we shall classify generally as "dithering methods". We think dithering is better at preserving fine details, such as the backscattered energy at about 1.5 seconds. Because this energy dips in a different direction than most of the other arrivals, and since it appears on only a few traces, it is very difficult to see on the wiggle trace display.

Every pixel in the dithered image carries amplitude information, unlike the wiggle trace display, where there are many unused pixels. It is not surprising that dithered plots can be significantly smaller than wiggle trace plots and still convey useful information. Figure

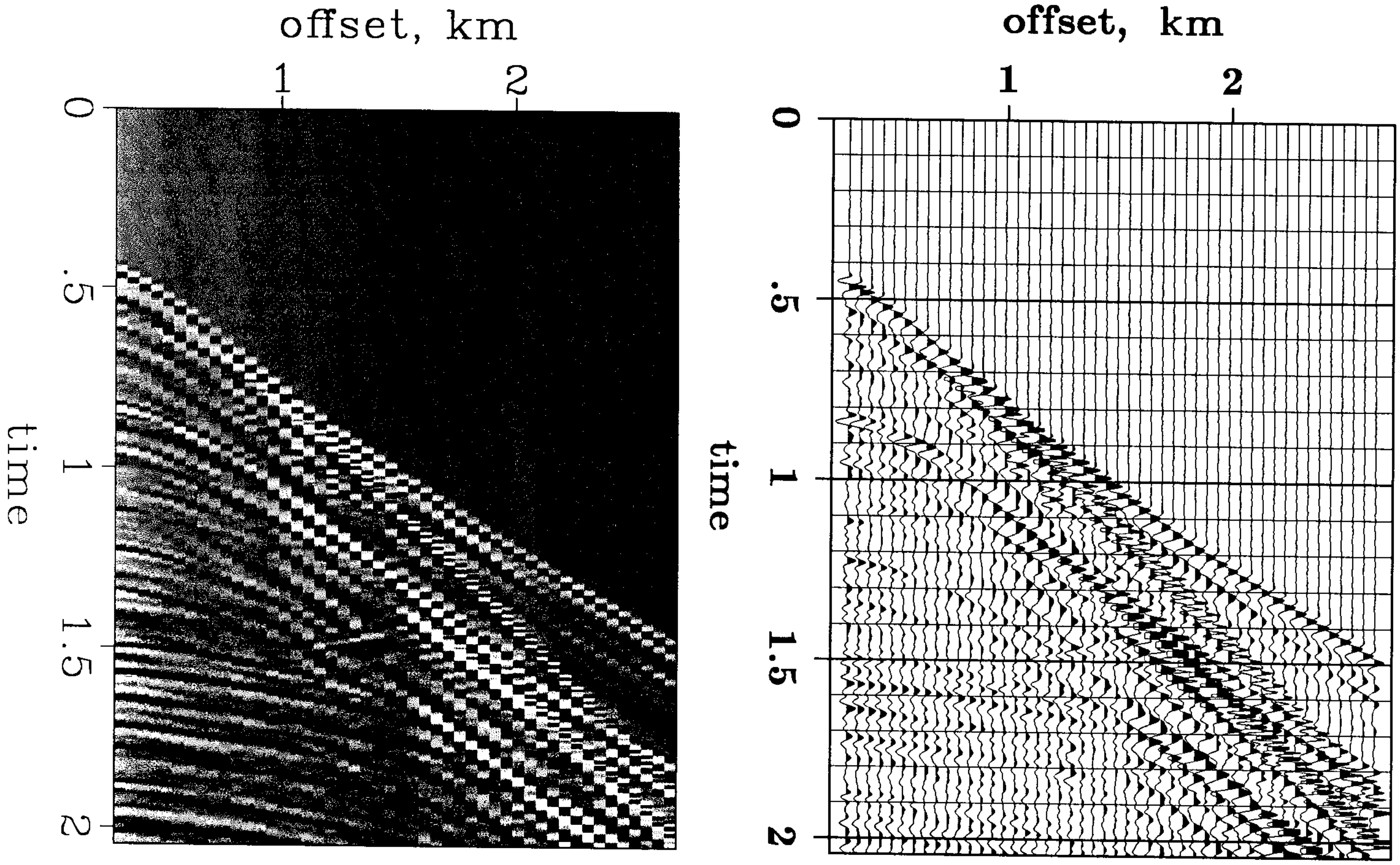


FIG. 11. Comparison of different plotting techniques. Note how much more visible the backscattered energy at 1.5 seconds is in the "grey" display than in the wiggle trace display. The Floyd-Steinberg method was used to convert the data to bilevel form.

12 shows an ambitious example, a combined display of CDP gathers and stacked traces that contains 845 traces, each with 1024 time samples. Admittedly it is difficult to discern subtle details at this scale. However, considering the volume of data and the size of the plot, we think the amount of detail present is surprisingly large. Note that for each data sample, there are approximately 2.6 pixels in the output image, enough to carry significant information despite the small size. We have not displayed a comparable wiggle trace plot, since we think that the wiggle trace method would have difficulty producing a reasonable image at 169 traces per inch and 1.25 inches per second!

Figure 13 shows the stacked section used in the test pattern in its entirety. There are several subtle features that show up well despite the size of the plot, such as the numerous diffractions and the subtle fault plane reflection located at approximately 3.5 kilometers and 2.2 seconds. The dataset contains 350 traces and 750 samples per trace, so the output

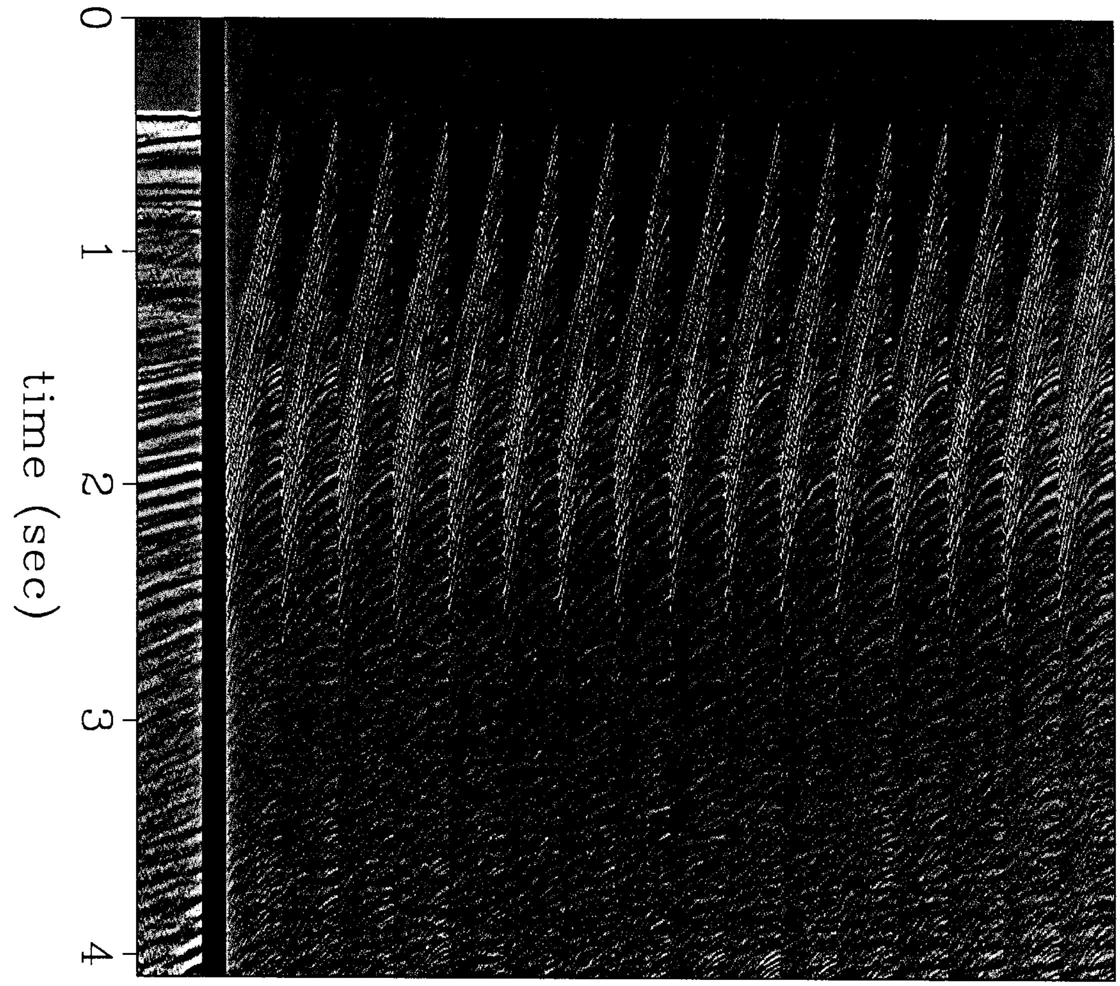


FIG. 12. Display of CDP gathers and the corresponding part of a stacked section. A wiggle trace plot would have to be much larger to generate a reasonable plot of a data set this size.

image contains 5.5 pixels for each input sample.

As a final example, Figure 14 shows a view of a three-dimensional cube of seismic data output from Rick Ottolini's movie program. The resolution is limited by the size of the original image — only 131 by 150 pixels. Compare to Figure 3.1-6 on page 159 of *Imaging the Earth's Interior.*

CONCLUSIONS

We have described a new method for plotting seismic data. The key step is converting the data from 8 or more bits per pixel of grey-scale color to single-bit bilevel form. If thresholds for this conversion are chosen properly, the conversion can be performed with no significant deterioration in image quality. Care must be taken to apply necessary corrections to account for the characteristics of the plotter.

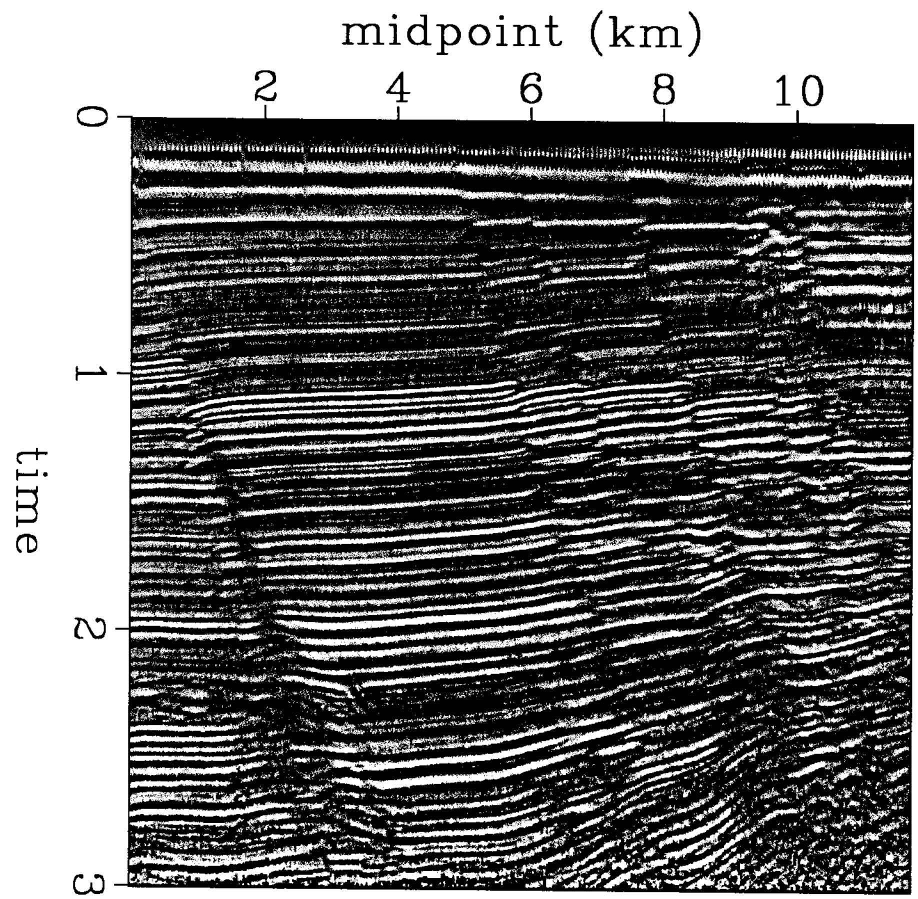


FIG. 13. Display of the stacked section used in the test pattern. The pattern of white dots near the top of the section is due to irregular offset sampling.

The dithering method of plotting seismic data offers some important advantages over the traditional wiggle trace display. Subtle features are often easier to see, especially if they occur over only a few traces or at other than the predominant dip direction. Plots can generally be much smaller, since all the space is used to display amplitude information. Also, since the dithering method does not draw a vertical line for each trace, it doesn't add artificial vertical features to the plot, unlike the wiggle trace method. These features often obscure subtle details in the data. This seems especially true on graphics displays.

The wiggle trace display still has obvious advantages. An interpreter trying to pick first breaks needs the wiggle trace display. Amplitude variations are easier to quantify when the wiggle trace display is used, since the excursion of wiggles can easily be measured. Frequency is easier to estimate when the zero crossings can be seen clearly.

The better display method is the one that best preserves the features that interest the observer. This can best be determined by experimentation. Note that several other papers

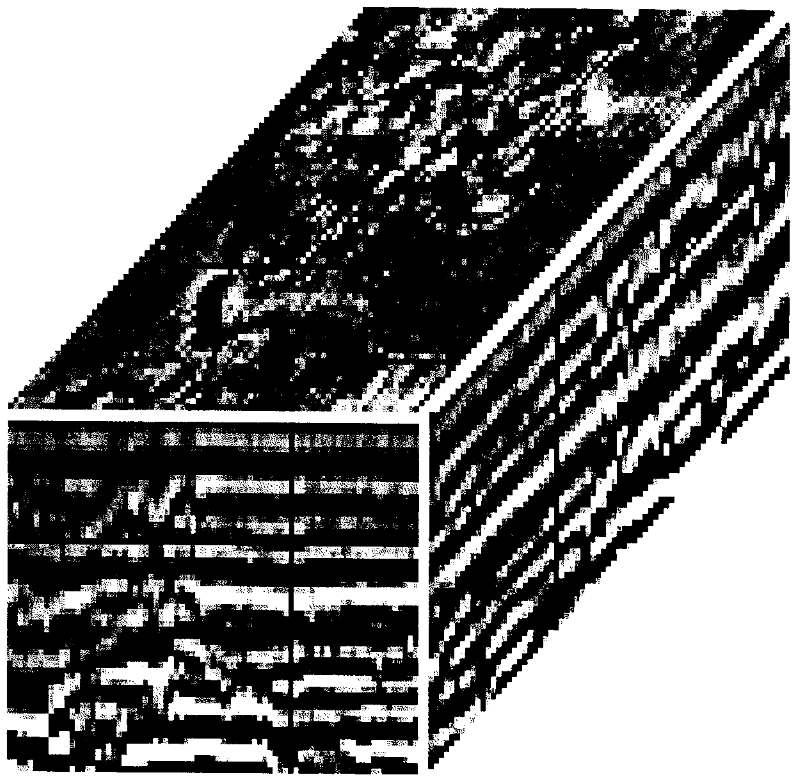


FIG. 14. Three-dimensional seismic data from Ottolini's seismic movie program.

in this report contain plots generated using the dithering methods described here, despite the fact that we were still working on the programs a few days before the report deadline! These additional examples should further help to define uses for these techniques.

A word about color

In this paper, we have limited the discussion to grey-scale or monochrome images. We think that these techniques can easily be extended to simulating color graphics displays on color plotters. Typically, colors are specified on such devices as combinations of three primary colors. If the plotter software is flexible, it should be possible to process the primary colors separately, using the techniques described in this paper, then combine them to create a full-color image. Alternatively, color plotting software (and perhaps some monochrome plotting software as well) often includes a pre-defined color table that essentially performs for the user the mixing of primary colors to generate any particular color. Such a color table can be thought of as a collection of dithering matrices that allow the primary colors to be combined in ways that do not generate distracting patterns. It remains to be seen whether such color tables are adequate for color dithering or if an adaptive method such as the Floyd-Steinberg algorithm would improve color dithering as well.

Future work

We would like to better quantify and explain the effects that required correction in order to bring displayed and plotted images into agreement. Another obvious step is to apply these techniques to color plotting. Finally, we believe that the ordered dither technique can be improved by using non-rectangular dithering "patterns". Square dithering matrices introduce artifacts that are aligned horizontally and vertically and are therefore very noticeable to the eye.

ACKNOWLEDGMENTS

Jon Claerbout first suggested the need to better reproduce graphics display screen images. Rick Ottolini applied our techniques to obtain hardcopies of Sun workstation display screens. Francis Muir encouraged us to implement the Floyd-Steinberg method, which turned out to be a significant advance. All three provided many helpful comments and suggestions during the development described here. John Etgen, Paul Fowler, Stew Levin, Jos van Trier, and Marta Woodward provided lots of useful feedback. Chuck Sword made a visible contribution to our project, with the help of Brian Quinn and his video equipment. Scott Douglass first suggested the use of dithering. Professors Robert Floyd and Donald Knuth of the computer science department responded to our inquiries about dithering with reprints of their papers on the subject and answered several questions.

REFERENCES

Claerbout, J.F., Imaging the Earth's Interior, 1985, Blackwell Scientific Publications.

Floyd, R. and Steinberg, L., 1975, An adaptive algorithm for spatial grey scale, SID 75 Digest, Society for Information Display.

Jarvis, J.F., Judice, C.N., and Ninke, W.H., 1976, A survey of techniques for the display of continuous tone pictures on bilevel displays, Computer Graphics and Image Processing.

Knuth, D.E., Digital halftones by dot diffusion, 1987, in press.

Ottolini, R., Sword, C., and Claerbout, J.F., On-line movies of reflection seismic data with description of a movie machine, 1984, Geophysics.

Pratt, W.K., Digital Image Processing, 1978, Wiley.

Sword, C., Claerbout, J.F., and Sleep, N., Finite-element propagation of acoustic waves on a spherical shell, SEP-50.