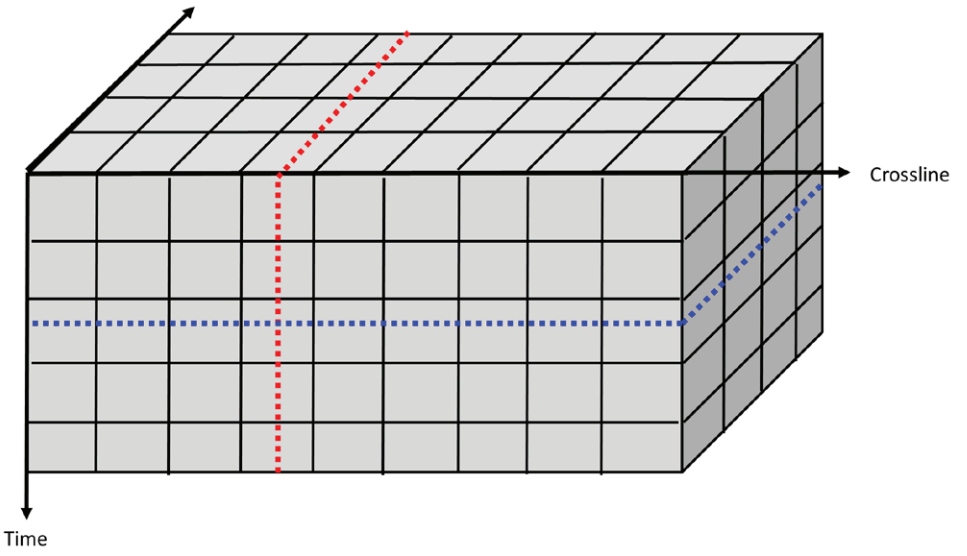
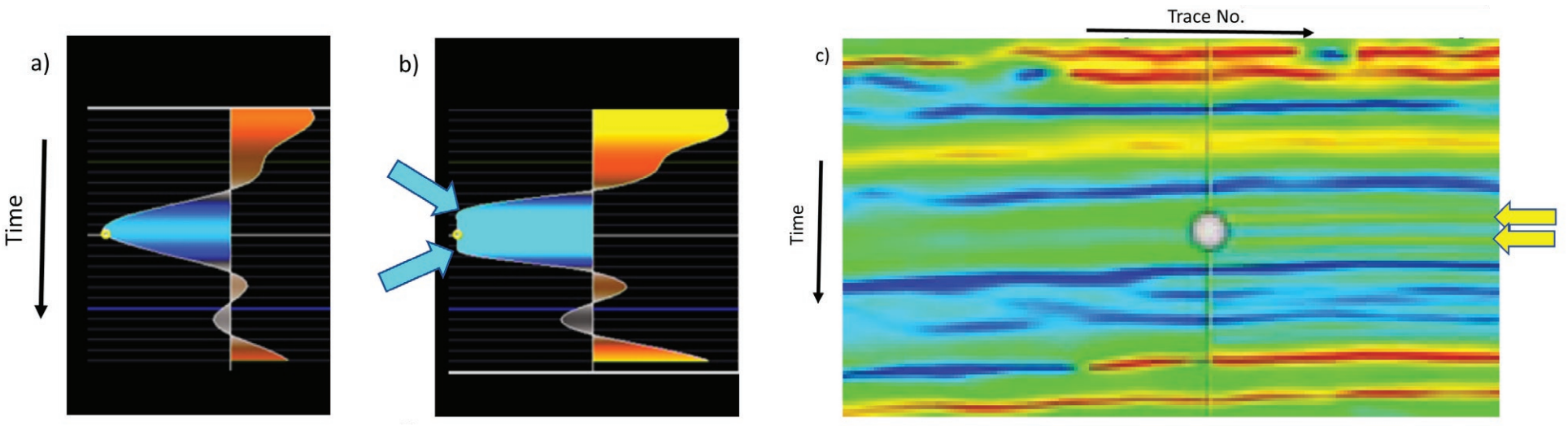
Geophysical Corner

Getting Bit by Too Few Bytes

Right: Figure 1. A cartoon of data stored as bricks common to most interpretation workstation software. Traditional line-by-line storage works well when accessing an inline vertical slice. Crossline slices are less efficient in that the disk head needs to be moved for each trace, reading in a block of traces stored on disk and rejecting most of them. Time slices are the most inefficient, requiring the entire data volume to be read into memory, extracting just one sample per trace and rejecting the rest. In contrast, brick storage has equal efficiency in reading inline, crossline, and time slices. In this cartoon, only 1/5 of the data would need to be read into memory in order to display the blue time slice. Further speed is provided by storing the data as either 8-bit or 16-bit integer format, though care must be taken in data scaling.



Below: Figure 2. Potential problems with improperly scaling the data before converting to 8-bit integer format. (a) A high-amplitude trace scaled using (a) eight times and (b) four times the RMS amplitude of the seismic survey. (c) Juxtaposed vertical slices through instantaneous frequency volumes computing using (left) the scaling in (a) and (right) the scaling in (b). Note that the peaks and troughs of (a) are well rounded but the trough of (b) has been clipped and is set to the minimum value (i.e., a scaled version of -127) that can be represented by an 8-bit integer. The corners in (b) indicated by the cyan arrows give rise to high frequency artifacts indicated by the yellow arrow in (c) computed using the improperly scaled data on the right of (c) that do not appear when using the properly scaled data on the left of (c). (After Sheffield and Payne, 2001.)



Data are commonly stored on hard disks as 4,096-byte sectors that represent 1,024 four-byte floating point seismic samples. Even if only one sample is needed to be read into memory, the entire sector is read in. Traditional SEGY (the standard file format of the Society of Exploration Geophysicists) 32-bit floating point data are stored trace-by-trace, inline-by-inline format, which provides rapid reading and writing of inline vertical sections, slower reading (because the disk head needs to be moved) of crossline sections, and very slow reading of time slices, in which case every trace needs to be read into memory, the desired sample extracted, and the remainder of the trace discarded. For reasons of efficient data access and display, almost all commercial interpretation software converts the 32-bit floating point data stored trace-by-trace in SEGY format to a more efficient brick format.

Examining figure 1, note that only the bricks containing the dotted blue time slice need be read into memory, resulting in significantly improved data access time. Common brick sizes measure 64 x 64 x 64 samples containing 10,878,976 bytes of data. If there are 1,024 samples per trace, the amount of data read into memory is reduced by a factor of 1,024/64=16. Further increases in data access and display can be obtained by converting the 32-bit seismic data to either 8-bit (1-byte) or 16-bit (2-byte) integer format, thereby reducing the time needed to read data from the disk but also increasing the number of (lower resolution) data samples that can be held in computer core memory. Using 16-bit integer format is much safer than using 8-bit integer format but can still give rise to the pitfalls discussed below.

Clipping the Amplitude Peaks and Troughs

In general, few interpreters (including the authors) can see the difference in the display of a properly sampled 8-bit seismic amplitude data volume using 256 colors. However, subsequent attribute calculations can be quite sensitive to clipping the data, which unfortunately

Continued on next page ▶

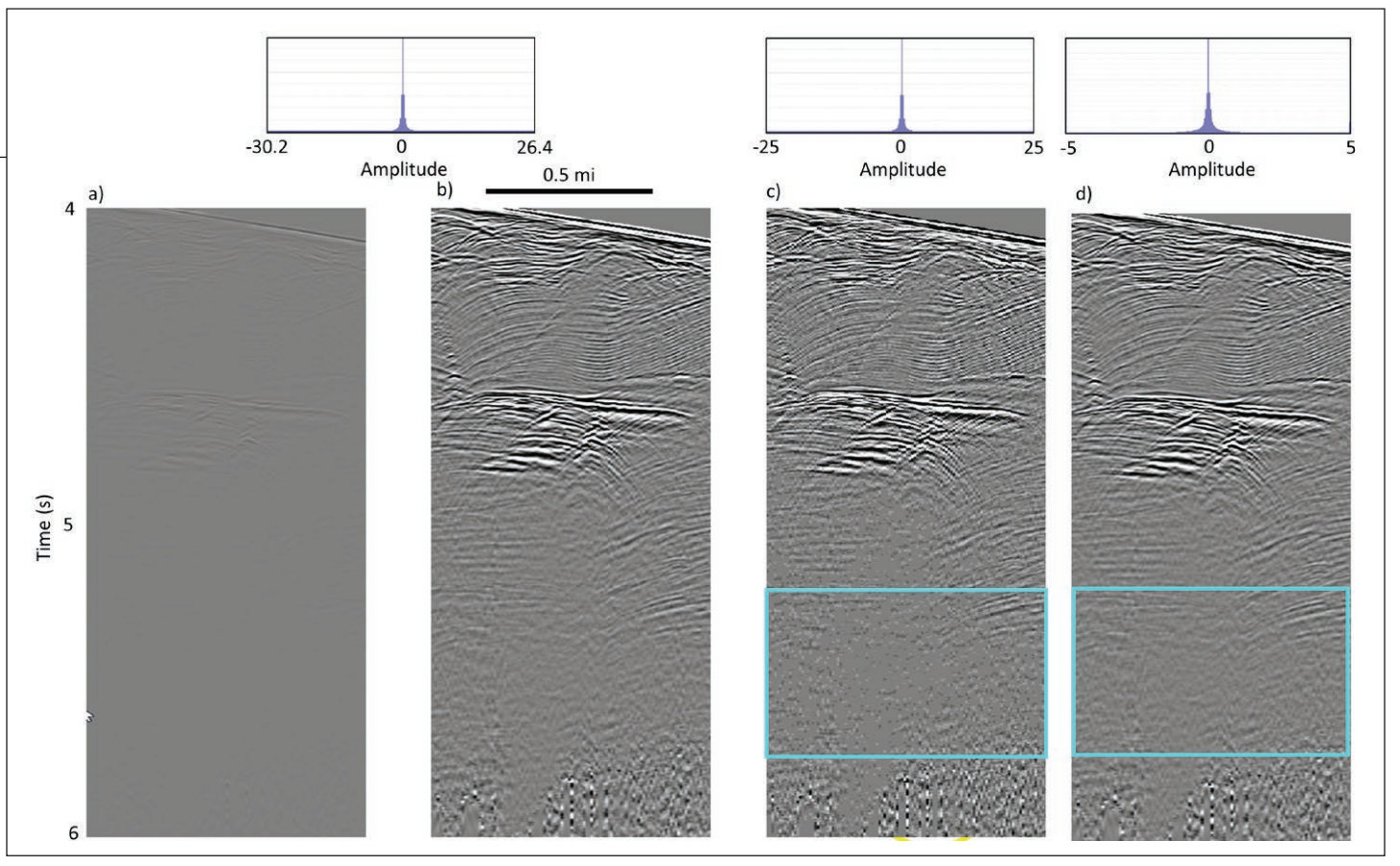
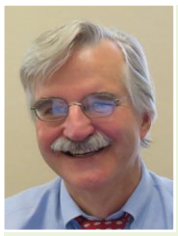


Figure 3. Limitations in representing data with large variations in amplitude using 8-bit integer format. This example is from a 3-D scientific survey acquired over Blake Ridge, offshore East Coast United States representative of many used in the academic community showing a segment of 32-bit data (a) as originally displayed by the commercial software with amplitudes ranging between -30.2 and +26.4 and (b) with a tighter display range of amplitudes ranging between -2 and +2. (c) . (c) The same data now converted to 8-bit integer format using balanced scaling parameters of -25 and +25, close to the range of the input and clipping less than 0.0001 percent of the data. Unfortunately, this scaling does not provide enough resolution for the low-amplitude reflectors within the cyan rectangle resulting in zero-value (pure gray) samples. (d) The same data converted to 8-bit integer using a more constrained amplitude range of -5 to +5, resulting in 0.12-percent clipping of the data at the low end and 0.14-percent percent clipping of the data at the high end. This resulting image is probably good enough for interpretation, but the values are "quantized" and may lead to inaccurate attributes in this low amplitude area.



Kurt Marfurt has divided his career nearly equally between industry and academia and is currently an emeritus professor of geophysics at the University of Oklahoma. For the past 28 years he has focused on seismic attributes and machine learning to aid the seismic interpreter. He has served as editor-in-chief for the SEG/AAPG journal Interpretation, has delivered two SEG distinguished instructor short courses, was the 2021-22 AAPG/SEG distinguished lecturer, and honored with AAPG's Robert R Berg award for research in 2019.

◀ Continued from previous page

happen around the high amplitude bright spots of interest. This pitfall has been illustrated before (shown in figure 2) as it negatively impacts the computation of instantaneous frequency. Here, the "corners" in the seismic wavelet introduced by clipping give rise to high frequency artifacts in the data that do not represent the underlying geology.

Representing Continuous Floating-Point Values as Quantized Integer Values

If we are careful not to clip the data, there might still be insufficient dynamic range in an 8-bit integer to represent subtle amplitude changes in the weaker parts of the data. In the simplest implementation (and a common default in interpretation workstation software), conversion to 8-bit data simply quantizes the data into 256 bins that span the range between the minimum and maximum range in the data. For the data in figure 3, the data range from -30.2 to +26.4. Using this range to also display the data against a gray scale color bar produces the image in figure 3a. Setting color bar to range between -1 and +1 increases the contrast in the image and gives the display in figure 3b. Next, we convert the 32-bit floating point data scaled to a slightly clipped, symmetrical range between -25 and +25 to 8-bit integer format, plot it against a color bar ranging between -1 and +1 and produce the result in figure 3c. Note that there are now a number of zero-valued samples within the cyan rectangle. Changing the limits of floating-point to integer range to be -5 to +5, clipping more of the extreme values, and plotting the data against a color bar ranging between -1 and +1 produces the image in figure 3d which appears to have fewer zeroes in its cyan rectangle.

Figure 4 zooms in on the data shown in the cyan rectangles shown in figure 3. In figure 4a note that there are only five values of amplitude displayed. Clicking each sample indicates these values are quantized to be ±0.39, ±0.20, and 0.00 which corresponds to an amplitude bin size of (25-(-25))/255=0.196. In contrast, the image in figure 4b shows a more continuous behavior because the bin size is smaller (5-(-5))/255=0.0392, or five times as many quantized values to span the range -1 to +1.

The negative impact of coarse quantization of continuous seismic data on subsequent attribute calculation (and even autopicking) can be significant. Figure 5 shows the instantaneous frequency data computed from the (figure 5a) original 32-bit floating point data, (figure 5b) the 8-bit integer data computed by scaling from -25 to +25, and (figure 5c) the 8-bit integer data computed by scaling from -5 to +5. Clearly, the results in figure 5b below t=4.8 s is erroneous and should not be used, whereas the results in figure 5c are comparable to

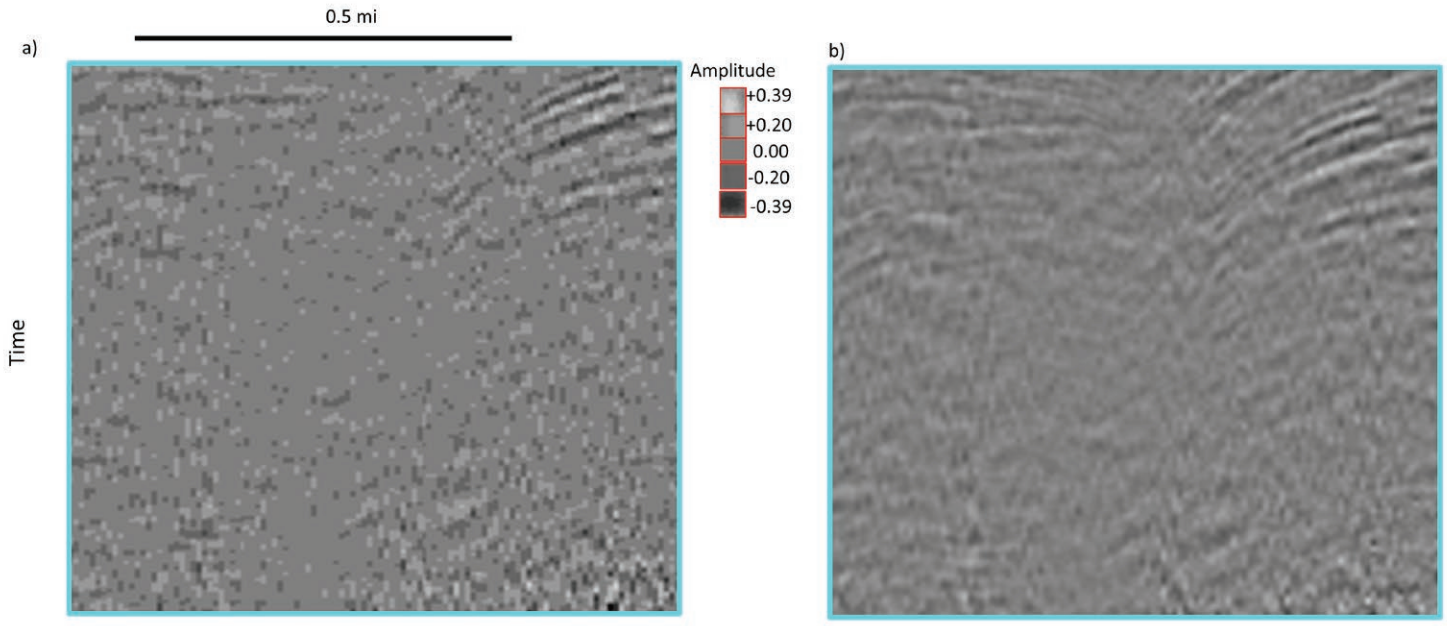


Figure 4. The quantization effect of expressing 32-bit floating point data as 8-bit integers showing the data in the cyan boxes of figure 3. (a) the 8-bit data scaled between -25 and +25 showing a highly quantized variation in amplitude, and (b) the 8-bit data scaled between -5 and +5 showing a smoother amplitude variation. Both images are displayed the same way, with the color bar ranging between -1 and +1. In (a) note that there are only five values seen with discrete amplitude read-outs of ±0.39, ±0.20, and 0.00. The data are no longer smooth and give rise to errors in subsequent attribute computation. By construction, the image in (a) has five times as many (still quantized) levels in this same area.

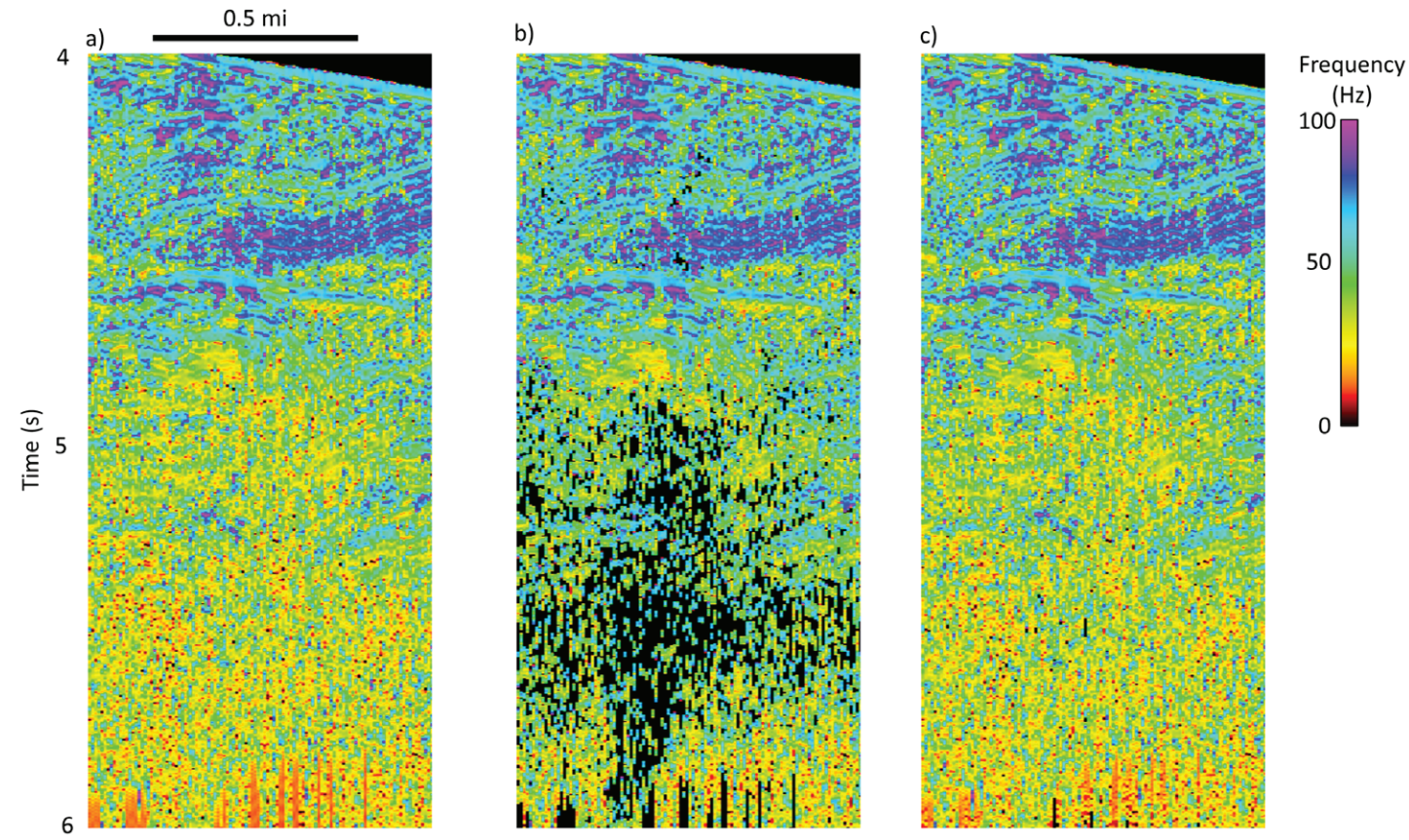


Figure 5. The effect of quantization by representing 32-bit floating point data as 8-bit integers. The instantaneous frequency computed from the (a) 32-bit floating point data shown in figure 3a and b, (b) 8-bit data scaled between -25 and +25 shown in Figure 3c, and (c) 8-bit data scaled between -5 and +5 shown in figure 3d. The results in (c) are full of errors in the low-amplitude part of the data, whereas the results in (c) contain a perhaps an acceptable number of errors, but also runs the risk of clipping the data in high-amplitude bright spots.

those computed from the original 32-bit data with only a few artifacts. Even with careful quality control, using 8-bit integer data as input to quantitative interpretation software such as impedance inversion can be disastrous. Given these observations, and the fact that almost all attribute computations are computed using 32-bit data, the authors advise interpreters to always keep a 32-bit version of their amplitude data in their interpretation project for subsequent attribute calculation, along with an 8-bit or 16-bit version that might provide faster interaction.

Potential of Changing the Value of Zero, Exporting an Imprecise Data Volume

No single software package does everything – some are better for interactive interpretation, some for processing, and some for quantitative interpretation. For this reason, it is common practice to export a given volume from the first software package into a simple 32-bit floating point SEGY standard format file and then import this file into the second software package. In addition to the pitfall of overly coarse quantization described in the previous paragraph, there is also the risk of changing the value of zero in the data, and hence of subsequent mute zones. In the software package used in figure 3, the default histogram ranges between -30.2 and +26.4. If the data are converted to 8-bit integers using these scaling parameters, the value of the samples in the water column no longer read as 0.0 but as -0.02. Furthermore, upon exporting this 8-bit data volume to 32-bit floating point SEGY format, the sample values in the water column (and in the "zeroed" values shown in figure 3a) are now equal to -0.235365.

Although these values are close to zero, their impact on subsequent attribute calculation can be disastrous. The brick formats used in many interpretation software packages do not retain the original trace headers containing the top and bottom mutes in the data. Fortunately, these can be calculated in a second software package by simply searching for the occurrence of the first and last non-zero value on each trace. Such a search will fail if the value of zero has been changed. An extreme case is AVO analysis where a curve is fit to represent non-zero far offset data rather realizing that these far-offset data do not exist.

The pitfalls discussed above occur most commonly when an interpretation "project" is provided to a partner or (in the case of the first author) student containing only an 8-bit version of the original amplitude data. Almost as common, the original interpreter "crops" an 8-bit subset of the data in their workstation software and exports it as 32-bit SEGY volume to be shared. In this latter case, there is no indication that the data were at one time only 8-bit. To avoid these potential pitfalls, always keep and when appropriate share the original 32-bit data.

(Editors Note: The Geophysical Corner is a regular column in the EXPLORER, edited by Satinder Chopra, founder and president of SamiGeo, Calgary, Canada, and a past AAPG-SEG Joint Distinguished Lecturer.)