Developing formats to handle large data set management and interpretation productivity

The massive size of data sets arising from current and emerging seismic acquisition technology poses a potential nightmare for the interpreter. John Kerr, Mark Lance and Janet Schweitzer of Landmark Graphics Corporation explain how the company is developing formats and compression strategies to tackle the problem.

There is little doubt that interpreters are facing ever-increasing volumes of seismic data, as E&P business models demand larger and larger survey sizes. Consider that only 10 years ago 3D surveys were primarily acquired over proven fields and used to optimize development programs. Today, interpreters routinely use 3D seismic to evaluate vast areas for regional studies, basin analysis, and lease sales. Service companies have dramatically increased acquisition capacity, resulting in cost-effective large-scale 3D surveys that cover thousands of square kilometres. Acquisition vessels that once towed a limited number of 240-channel streamers are now built and equipped to tow up to 20 streamers, between 6000 and 8000 meters in length. Large, high-resolution surveys have become the norm.

While individual seismic data volume sizes are increasing, so are the number of attribute volumes used for detailed interpretation and analysis. The use of attribute volumes like similarity, dip-azimuth, impedance, and offset stacks, to name a few, can quickly introduce a 10-fold increase in the number of full seismic volumes for interpretation.

Furthermore, as 4D seismic is utilized more often to monitor reservoir performance, and as the industry embraces multicomponent data to better image the subsurface or characterize the reservoir, interpreters will undoubtedly face even greater volumes of data.

While improvements in the geocomputing environment, like CPU performance, memory, disk capacity, and networks, are addressing data volume issues, interpretation software is also being developed to meet the performance needs and expectations of oil and gas company geoscientists. For example, Landmark has been introducing new horizon and data formats designed to improve workflow productivity where large data sets are concerned.

New horizon and seismic data formats

In mature fields, there are often hundreds or even thousands of horizon files that need to be stored on disk. As surveys are merged and survey sizes have grown, so too have the sizes of corresponding horizon files. With large data sets, the horizon tiling option dramatically improves overall display speed and conserves disk space.

Two new 3D seismic data formats, bricked and compressed, have also been introduced in Landmark's Interpret2000 software release. These data format options are intended to provide additional flexibility to address the data management, data access, collaborative and conventional workflow productivity issues that interpreters face daily with large seismic data sets. They also offer options to preserve data fidelity far beyond that of traditional 8-bit .3dv files.

The tiled horizon file format addresses storage and display performance issues for large data sets. It conserves disk space by storing only those tiles where interpreted data exists. Horizon display speed is improved by indexing tiles and accessing only horizon data that has been interpreted.

Previously, interpreted horizon data would be stored in a file that pre-allocated the space necessary to store a value for every trace on every line in the active 3D seismic project, whether or not a horizon pick existed. All values were arranged consecutively in an in-line orientation, resulting in relatively quick in-line displays, but cross-lines and arbitrary lines took longer to display.

In small to medium sized surveys, the difference in display times generally went unnoticed. However, as survey sizes became larger, display times for cross-lines and arbitrary lines increased significantly.

The tiled horizon format stores horizon picks in tiles having preset in-line and cross-line dimensions. No pre-allocation of disk space occurs, so the horizon file size is a function of how much data has been interpreted as opposed to the total number of lines and traces in the seismic project. The file grows as the horizon is interpreted, in many cases saving considerable disk space. Overall display performance is improved because less data needs to be read.

The bricked file format stores seismic data as three-dimensional bricks. Each brick contains data for a user-specified

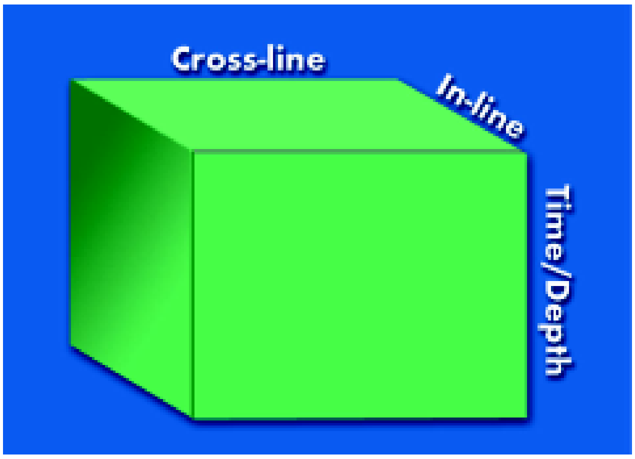

Figure 1

number of cross-lines, in-lines and time or depth samples. This allows interpreters to optimize data display performance for large data sets by specifying brick dimensions and orientation. They have the flexibility to design brick dimensions that match workflow needs and preferences (Fig. 1). These improvements have been achieved by creating a file that reduces the number of disk seeks and reads that are necessary to obtain the requested data. Data is indexed so that only the data necessary to construct the desired view is accessed and delivered to the interpreter's workstation.

In the past, trace data used to be written as a .3dv format file. This typically optimized display performance for the in-line direction only. As a result, interpreters often constructed an additional volume optimized for interpreting cross-line views and another volume to provide time slice or depth slice oriented views. Multiple volumes of the same data were required to optimize display performance for each vertical view, plus another volume for the horizontal view. This type of workflow procedure continues to be optimal for small to medium size (under 15 gigabytes) data sets. However, for very large data sets, managing multiple volumes may be problematic in terms of both data storage and workflow performance.

Where interpretation workflows involve large data sets, 20 gigabytes and larger, bricked files can be designed to optimize performance for a particular display orientation, similar to .3dv files (Fig. 2). The bricks can also be designed to provide faster overall display performance where random in-line, cross-line and arbitrary line views are routinely needed in the workflow.

Landmark has also worked on giving interpreters greater flexibility in selecting output sample formats for storing data. The bricked formats can be stored in one of five output sample bit formats. In addition to the 32-bit floating point format, new 8-bit floating point and 16-bit floating point formats are offered. Each format preserves the amplitude range of the original data far more accurately than clipping and scaling to an 8-bit or 16-bit integer values.

Compression issues

Landmark's proprietary compression algorithm (patent pending) is a lossy, JPEG-like process adapted for 3D volumes of seismic data. It compresses small overlapping blocks of the

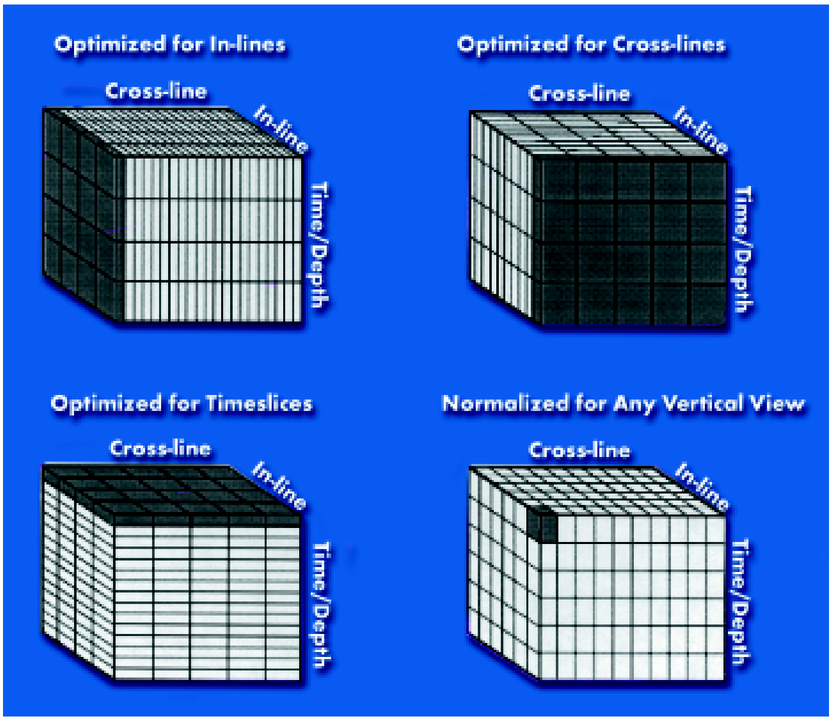

Figure 2

input volume with special care to avoid artefacts during compression. No clipping or scaling is performed when data is loaded. This means that maximum dynamic range is preserved and data displayed using 32-bit floating point values.

Faster computing platforms, improved communication networks and data storage technology have all contributed to accommodating the seismic data explosion. Interpreters, however, continue to face workflow performance challenges on large data sets. In an effort to maintain productivity, interpreters routinely adapt their data and workflows to accommodate hardware constraints, legacy architecture, disk space limitations, network bottlenecks and time demands.

Historically, interpreters have adopted data decimation strategies to save time or to adapt to computing environment limitations out of their control. In an attempt to manage increasingly large volumes of seismic data, geoscientists have routinely reduced the size of the data volume by summing adjacent traces before processing, by storing data in 8-bit formats, or by simply throwing out traces or lines so that the remaining data fits within limitations of the interpretation environment. Data was also often divided into small sub-volume data sets for interactive viewing and manipulation. This strategy has been especially prevalent where mega-marine surveys have continually increased the upper bounds for seismic survey size.

Decimation strategies lead to 2:1 disk storage savings in the case of trace summing, 4:1 savings in the case of 8-bit truncation and 2:1 savings in the case of 16-bit integer storage. However, these benefits come at the expense of throwing out valuable information, loss of dynamic range, or at the risk of losing regional context while viewing small subsets of data, all which may have a negative impact on interpretation decisions.

Data decimation strategies have minimal positive impact on IT resources and have a negative impact on interpretability when compared with strategies that leverage data compression technology. Although interpreters will always face IT resource and time constraints, compression technology provides additional flexibility in developing strategies for productive.

Data compression can offer much greater disk savings than traditional decimation strategies. For many data sets, compression ratios of 20:1 may become routine while greater ratios could be suitable for some interpretation workflows. Also, consider the immediate benefit of compression technology on data archiving. Compressing 100 gigabytes of data to merely 10 gigabytes, a 10:1 compression ratio, provides direct savings in disk storage costs. Storing and managing seismic or attribute volumes on local systems also becomes practical for much larger data sets.

By leveraging compression technology, interpreters will be able to merge multiple surveys into a single project and store much larger surveys on the local workstation. This may well improve interpretation productivity in basin-wide or regional evaluations. For example, access to data from mega-surveys, as a single project, will enhance workflow productivity for lease sale evaluations where large volumes of data often have to be interpreted quickly.

While localized IT benefits are clear, compression technology can also dramatically reduce network data traffic in a distributed environment. Compressed data sets are much more portable than their .3dv file format equivalent. Data sets that would be impractical to transfer over the Internet or a WAN in uncompressed form may be reasonable to transfer in compressed form.

Using compression technology to reduce disk space requirements will also allow the interpreter to store multiple volumes of the data. Access to such multiple volumes can enhance workflow productivity when detailed analysis is required to support drilling decisions or production facilities planning.

For example, in the case of a highly faulted Tertiary prospect where AVO anomalies indicate productive zones, volumes from several different processing sequences may be required to unravel the complex geology and increase confidence that hydrocarbons are present. A compressed amplitude volume could be used to describe the initial structural framework of the area and indicate amplitude anomalies. Additional attribute volumes provide sources of information that can improve confidence in the interpretation. A similarity volume, for instance, can show detailed faulting and compartmentalisation, and a waveform analysis volume may illuminate the stratigraphic or depositional framework. Near and far-angle stack volumes can give confidence in the presence of an AVO anomaly, while various other attribute volumes can help in reservoir characterization.

Compression technology should have significant impact on data storage and workflow productivity as 4D and multicomponent surveys bring more data to the interpreter's desktop. It will also play a role in streamlining access to prestack seismic data, the source of detailed velocity and lithologic information, during interpretation sessions. The benefits of the compression format go beyond reducing data size. Compressed data preserves amplitude information, because no clipping or scaling is used when the data is loaded.

In some tests working with data sets larger than 20 gigabytes, the average display speed when taken across inline, cross-line, arbitrary, and horizontal display orientations is improved over .3dv files or bricked files. This observation is related to the two sources of overhead for displaying seismic data, I/O and computation time. For .3dv files and bricked files, the I/O overhead is high relative to computational overhead. For compressed files, the I/O overhead is small relative to the computational overhead associated with dynamic decompression of the data.

When the data sets are small, the I/O overhead of 3dv or brick data set is less than the computational overhead of the

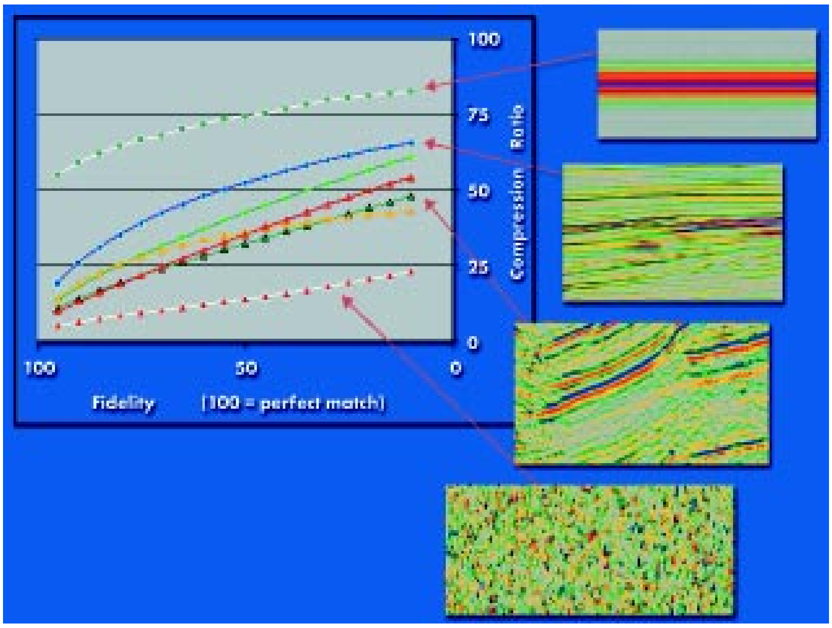

Figure 3

equivalent compressed file. For large data sets, this relationship changes, and the I/O overhead of a 3dv or brick data set is higher than the computational overhead of the equivalent compressed data set. Generally, the larger the data set size the greater the improvement in overall display performance.

Another advantage of the compressed volume format is that to access any vertical or horizontal view of the data, only one volume is required. This becomes significant where, for example, arbitrary views are required for interpreting a view perpendicular to a critical fault plane. This is often the case where large regional surveys cannot be acquired perpendicular to all areas of interest or where the subsurface contains complex faults. Display speed can be further optimised by taking advantage of client-side cache and by utilizing a unique performance option. Based on the pattern of requests, the next seismic line can be retrieved while the current section is being interpreted.

The data fidelity dilemma

Although data compression offers many benefits to interpretation workflow productivity and IT resource management, interpreters must balance these benefits with the inherent loss in data fidelity during the compression process. The notion of *fit for purpose* guides the interpreter's decision of how much fidelity loss is tolerable relative to the benefits gained from compressing the data set.

Since compression is lossy, that is, it does not preserve 100% of the input data, interpreters should first consider the degree of fidelity suitable or fit for purpose, to achieve the interpretation goals. For example, regional seismic evaluation may require less data fidelity than a reservoir characterization project. For rapid access to very large data sets an interpreter would be willing to tolerate less detail, or fidelity, in the data.

Landmark's data compression software allows the interpreter to specify the minimum acceptable fidelity of the output data when a compressed volume is created. Why not specify the desired compression ratio directly? One reason is that the relationship between preserved fidelity and compression ratio is different for every compressed block in a data set. For each block, compression 'adapts' to the quality of the input data and therefore typically varies throughout the seismic volume. The fidelity factor controls the amount of compression by specifying how closely the compressed data must match the original data, therefore a single fidelity factor, will equally preserve discontinuous or poor signal-to-noise ratio data deep in the seismic section as well as continuous high quality data in the shallow section.

The equation:

$$\text{Fidelity} = (1 - (\text{RMS Error})/(\text{RMS Original})) \times 100$$

specifies the acceptable ratio of RMS (root mean square) amplitude error to original RMS signal, introduced by compression.

Fidelity factors are specified by values from 1 to 99, where 99 preserves the highest degree of fidelity. The higher the fidelity factor, the lower the RMS error. Given the ability to control the fidelity of seismic data after compression, interpreters will be able to balance *fit for purpose* compression with interpretation objectives and data storage constraints.

Selecting the best fidelity factor for compression requires consideration of many factors including data characteristics, interpretation objectives and system configuration. To achieve the optimum, the relevant factors should be consid-

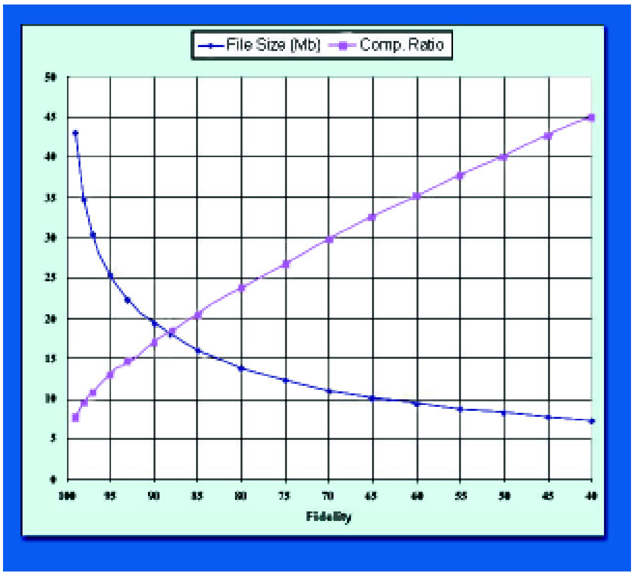

Figure 4

ered on a site-by-site and project-by-project basis. It is not possible to provide a set of rules for selecting compression ratios.

Figure 3 shows the relationships between fidelity and compression ratios for six 32-bit data sets from surveys of basins around the world, each exhibiting varying degrees of continuity and signal-to-noise ratios. For comparison, fidelity and compression values from a random noise data set and a synthetic wavelet data set are provided. As shown, for any given fidelity factor the achievable compression ratio differs for each data set. The achievable compression ratio is data dependent with the outer bounds of compression defined by the two synthetic data sets. As expected, the higher the fidelity value, the lower the compression ratio. Also, higher compression ratios can be achieved from more coherent data (refer to the curve for the single bandpass wavelet).

Figure 4 illustrates how compression ratios and data set size responded to varying input fidelity values in a high-quality data example. Here, when compressing the data using a fidelity value of 99 the original 32-bit data set size decreased from 328 Mb to about 43 Mb. This represents a compression ratio and disk saving of about 8:1. Using a fidelity factor of 95, the original file size decreased from 328 Mb to 25 Mb, a compression ratio of about 13:1.

Observing the file size curve, it is clear that the greatest rate of change in compressed file size occurs between fidelity factors of 99 and 90. This observation has great significance for interpreters. It shows that, for good seismic data, high compression ratios are achievable with minimal loss of fidelity.

The vertical and time slice examples in Fig. 5 suggest that for visual regional interpretation of high-quality data, interpreters can consider working with highly compressed, lower fidelity data. With data having a 48:1 compression ratio, major fault breaks and horizons in the seismic section are maintained, although some localized changes in background amplitude can be detected.

The display sections in Fig. 6 show the results of compression using a range of fidelity values. Note the compression ratios and the minor fidelity changes from one display to the next. Figure 7 is a zoomed-in view of a fault zone from the compressed data above. Note the minor loss of continuity, increase in background noise and changes in the short wavelength features between the original 32-bit data and the data after 40:1 compression.

In the zoomed examples the compressed data image (fidelity value of 50) exhibits minor but observable differences in the detail of the fault zone. However, the fault plane can still be placed with confidence in the middle of the disturbed zone. For this data set, fidelity values between 80 and 50 will pro-

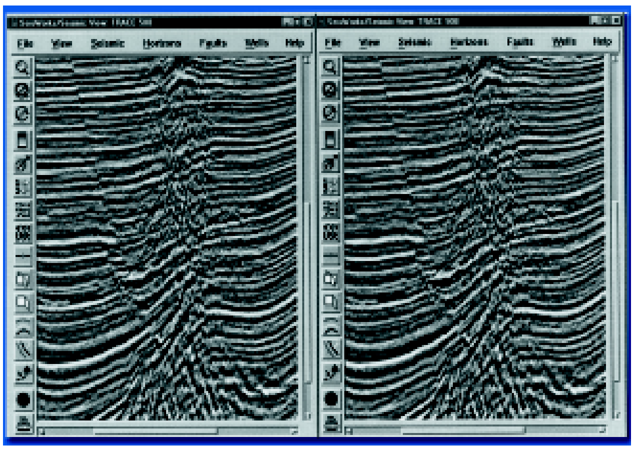

Figure 5

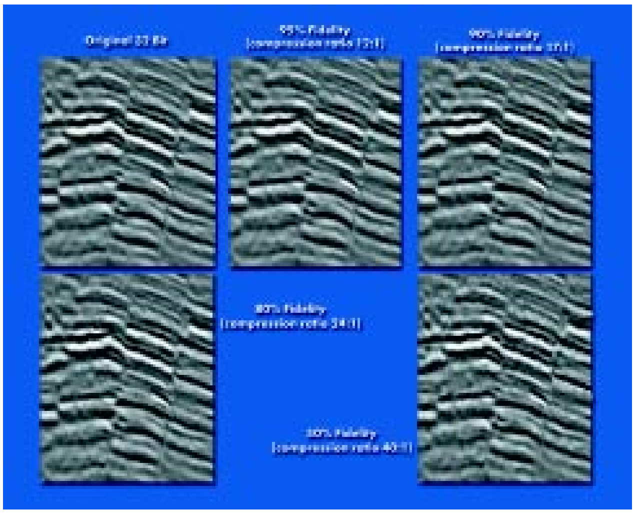

Figure 6

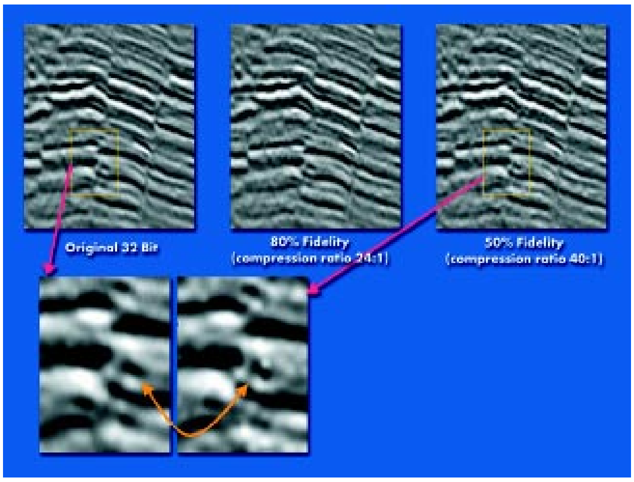

Figure 7

vide clear advantages in terms of disk usage and yield seismic data fit for regional fault and horizon interpretation.

Attribute volumes

Can attribute volumes be compressed? Seismic attributes are subject to similar generalizations as seismic data. Compression quality and achievable compression ratios depend largely on continuity and coherence in the attribute data. Most attribute volumes are well suited to take advantage of compression. Figure 8 shows original input data (top), instantaneous frequency attribute data (middle), and the compressed instantaneous frequency data using a fidelity of 90.

Questions arise regarding compressed data quality and how much fidelity is needed for a given interpretation task. Visual and measured differences between the original data and the compressed data can help establish *fit for purpose* criteria. The examples in Fig. 8 provide a visual comparison between original and compressed data. Often, from a visual comparison alone, interpreters can determine if the compressed data is *fit for the purpose* of regional or structural interpretation. Data set size, disk limitations and other factors will also play a part in the decision, but visually determining whether a compressed data set would materially alter an interpretation is a good place to begin. One might also consider more quantitative methods for measuring the difference between original and compressed data (Fig. 9). Landmark in fact provides tools for mathematically evaluating compression results.

Horizon pick times can provide a metric to analyze the difference between original and compressed data sets. The graphs show seismic horizon time pick differences between compressed data and original data. For comparison, picking differences for 8-bit truncated data is also provided.

Using auto-tracking, picks were recorded for a high amplitude trough, a low amplitude trough and a zero crossing from an original 32-bit 3D data set. These picks were then snapped to an 8-bit truncated volume and to several compressed seismic volumes. The difference in pick times between the original 32-bit data and the 8-bit and the compressed data is compared in a series of graphs (Figs 10–13).

Each graph represents data compressed with a different fidelity value. Figure 10 shows that pick times from 8-bit data (4:1 compression) and picks from data compressed using a fi-

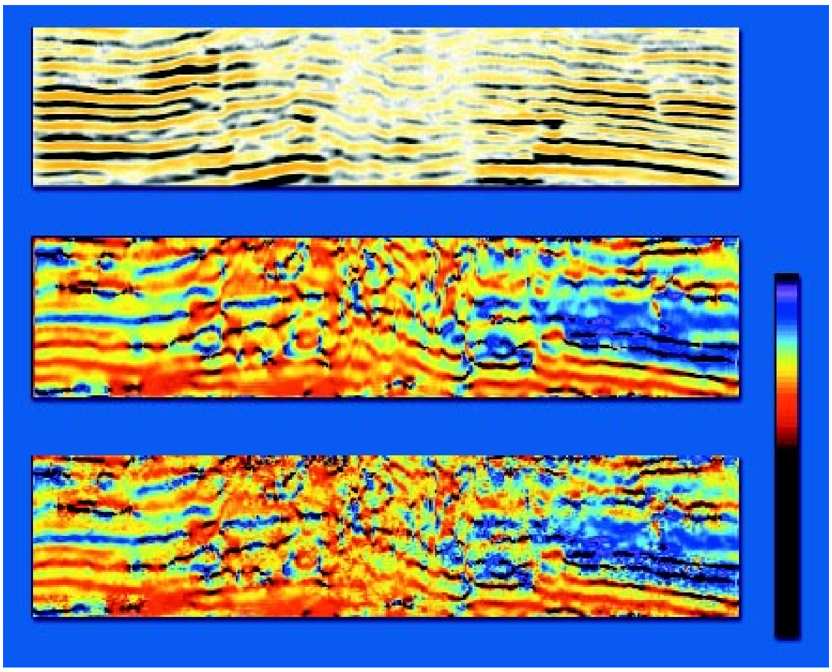

Figure 8

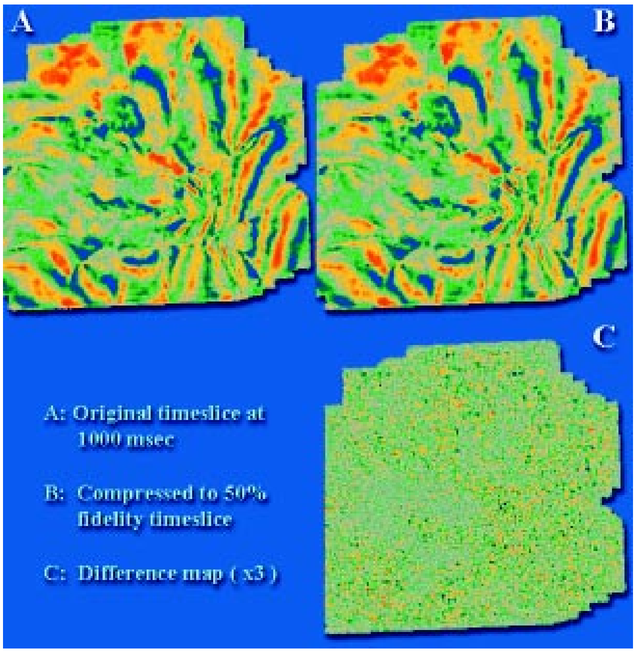

Figure 9

delity value of 99 (7:1 compression) are virtually identical. In Figs 11–13 the pick difference curves for compressed data begin to flatten, that is, more picks approach a difference of 2 msec as fidelity values decrease. It is interesting to note that the standard deviation of pick differences, even from the data compressed using a fidelity factor of 80 (compression ratio of 22:1), is less than 1 msec.

Selecting the optimum fidelity factor for any given data set or interpretation task will most likely require some trial and error. Observations and guidelines in this paper provide a starting point.

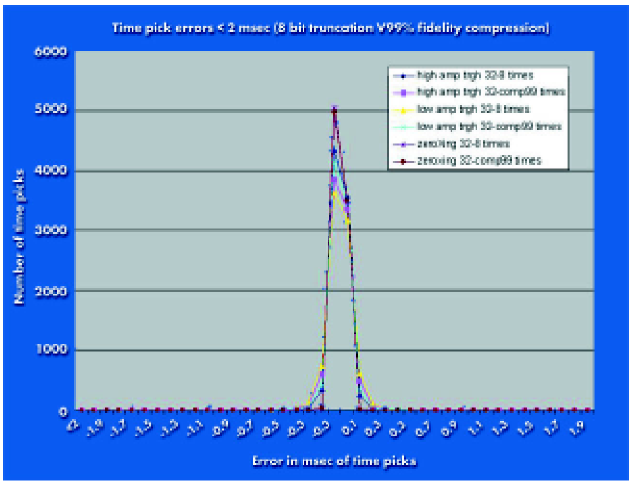

Figure 10

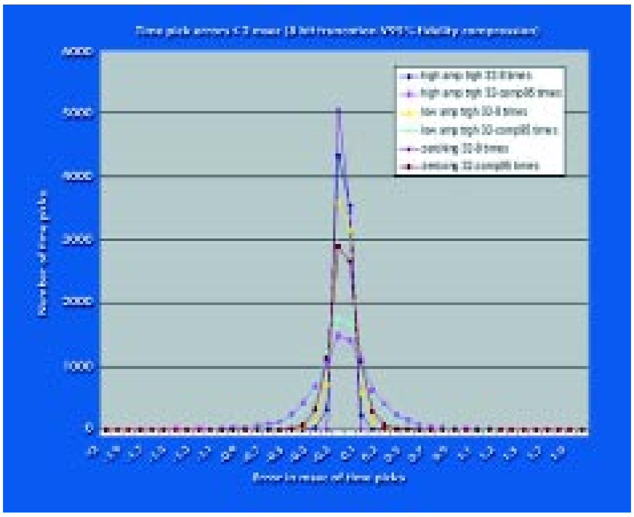

Figure 11

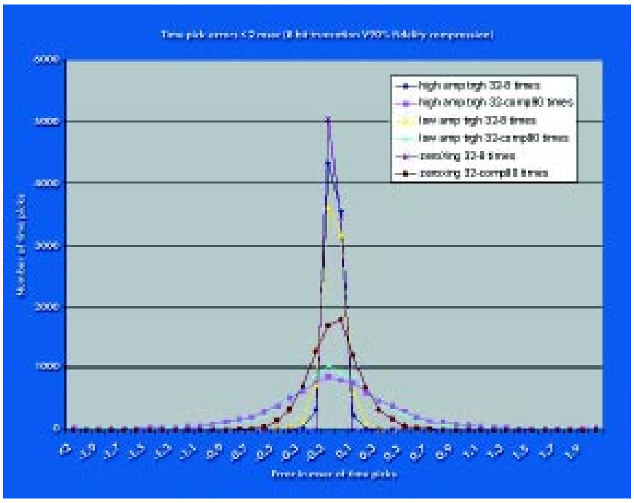

Figure 12

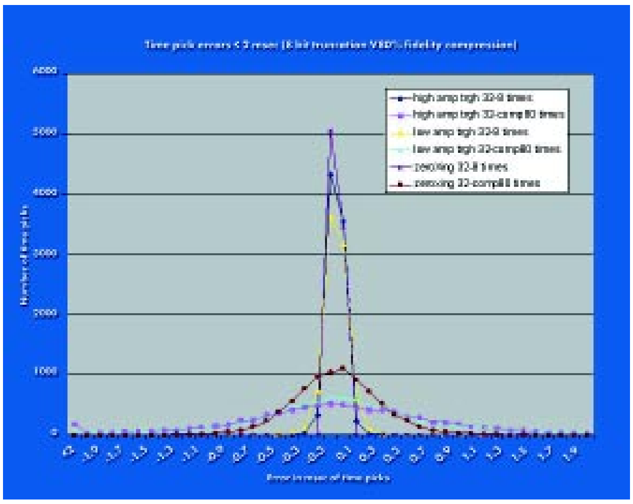

Figure 13