

2D Visualization of Seismic Trace Data Based on SEGY Format

Wenqin Li

*School of Computer Engineering, Yangtze Normal University, 408100, China
lwq912@126.com*

Abstract

Visualization technology now has been widely used in the field of seismic exploration, which can improve the quality and efficiency of data processing, reduce risk and cost of exploration. This paper mainly studies the visualization of 2D seismic trace data, and figures out the oscillograph of trace data based on the actual data. In the drawing of the oscillograph, to avoid the situation that oscillograph becomes polygonal line when dealing with fewer sampling points, this paper introduces Bezier Curve Fitting method, and as the test data demonstrate, the oscillograph becomes more smooth with the help of Bezier Curve Fitting, and the expected effects are achieved.

Keywords: *Seismic trace data, 2D visualization, Oscillograph Drawing, Bezier Curve Fitting*

1. Introduction

With the continuous improvement of seismic exploration technology, the seismic exploration data via field acquisition, with the help of the acquisition instrument, can be transformed into SEGY data format arranged in trace sequence directly [1]. The seismic data in SEGY format is stored in binary, and for common people, the trace header information and amplitude data value of the seismic data can not be observed easily. Besides, visualization technology has been broadly applied in the processing and interpretation of seismic data: on the one hand, it can free the geophysical prospectors and interpretation staffs from the heavy work of manual numerical reading, calculation, and drawing; on the other hand, it can greatly shorten the period for seismic data interpretation, speed up the process of interpretation, and thus achieve high speed and efficiency [2]. 2D seismic data visualization is not only the basis of seismic data interpretation, but also the basis of three-dimensional visualization. 2D visualization can improve the efficiency, precision and integrity of seismic interpretation. With the aid of 2D graphs, users can perform complicated transformation on their interested seismic data, and can mark the site they are interested in, which greatly improves the interpretation staff's interpretation degree on the dimensional visualization seismic data. Seismic exploration data are composed of trace headers and a number of traces with certain lengths. Trace header contains limited values and is used to provide the possible inter-trace change information and procession, trace recognition and the like basic information; trace data follow each trace header, and are actually measured seismic data. The paper mainly takes the visualization of trace data as the starting point, describes the drawing process of 2D visualization in combination with actual data, and finally, as per the requirements of seismic trace data display, raises the algorithms and strategies of curve fitting.

2. SEGY Seismic Data Format

SEGY seismic data format is an exchange format standard formulated and recommended by US Society of Exploration Geophysicists, and has become uniform

seismic data format in the world. The purpose of the format is that all the physical geography companies can share and exchange the seismic data they acquire and save calculation resources.

Understanding the document structure of SEGY format seismic data and correctly reading the seismic data is the premise to perform seismic data processing, so the deviation of any parameter can lead to the fault of the SEGY data documents and thus the data will not be interpreted. Standard SEGY document structure is composed of file header and data cubes, as is illustrated in Table 1. The file header occupies 3,600 bytes in total, which is divided into two parts: Part One is a head of string segment of 3,200 bytes which adopts EBCDIC codes to record the relevant information of the seismic acquisition system. Part Two is binary head segment of 400 bytes recording the data cube information, such as the sampling number, sampling rate, sampling format *etc.* The data cube is composed of a plurality of data traces, and each trace of data includes the trace header and trace data. Header is a binary data of 240 bytes, which records the sampling number, sampling interval, common depth acquisition point number, XLine No, YLine No and coordinate information, *etc.*

Table 1. Standard SEGY Document Structure

File Header		Data Cube				
3,200 byte string segment file header	400 byte binary file header	1st trace header (240 byte)	1st trace seismic data	nth trace header (240 byte)	nth trace seismic data

Generally speaking, original SEGY format seismic data are stored in the storage format of work station, and are different from the storage format on computer. For the binary data in work station, the high byte is followed by the low byte, namely BigEnd ina, while for the binary data in computer, the low byte is followed by the high byte, namely, LittleEnd ina. Therefore, to achieve the reading of SEGY format data under Windows background in computer, the SEGY data format in work station shall first be transformed into the SEGY data format which can be recognized by computer, namely performing the conversion of high and low bytes in reading. As per the different byte number occupied by the storage information, specific implementation codes are given, i2[] is the array for storing 16 digit data, i4[]is the array for storing 32 digit data, and n is the data number.

16 digit binary data conversion

```
Void swapbytes2(unsigned short i2[],long n)
```

```
{
# ifndef SUN_OS // Work Station Type
Long i;
Unsigned short b1, b2;
For (i=0;i<n;++i)
{
b1=i2[i]<<&;
b2=i2[i]>>&;
i2[i]=b1 | b2;
}
```

```

} // High and low byte conversion
# end if
}
32 digit binary data conversion
Void swapbytes4(unsigned long i4[],long n)
#ifdef SUN_OS // Work Station Type
Long i;
Unsigned long b1,b2,b3,b4;
Unsigned long i1,i2;
For (i=0,i<n,++i)
{
b1=i4[i]<<24;
b2=i4[i]>>&;
b2=b2<<24;
b2=b2>>&;
b3=i4[i]<<&;
b3=b3>>24;
b3=b3<<&;
b4=i4[i]>>24;
i1=b1 | b2;
i2=b3 | b4;
i4[i]=i1 | i2;
} // High and low byte conversion
# end if
}

```

3. Oscillograph Drawing

Seismic trace data are sampled from the waveform of the seismic signals by certain intervals, and then the series of amplitude values are recorded by certain manner. Oscillograph illustrates the seismic data of various traces within a plane in the form of curve, and demonstrates the vibration from the balance position of the reflection wave at the inspection spot. Oscillograph is used to express the vibration at the geophone sites with the manner of curves, namely the waveform of the geophone sites. Oscillograph can clearly reflect the dynamical characteristics of reflected waves (such as amplitude, frequency and waveform). The distances of the graphics region between the left and right boarders of the client region are s_x and s_y respectively. The trace data are stored in array `one_trace_data`, and then the horizontal coordinate position of the j th sampling site of the i^{th} trace is $X=s_x+i*n \text{ TraceInterval}+one_trace_data[j]$, and the vertical coordinate position is $Y=s_y+t*n \text{ TimeInterval}$, the horizontal coordinate position of the next sampling site is $X= s_x+i*n \text{ TraceInterval}+one_trace_data[j+1]$, and the vertical coordinate position is $Y= s_y+(t+1)*n \text{ Time Interval}$. One trace of data is read as per the starting time and

ending time, and after the coordinate (X, Y) is determined, the points are connected, thus the Oscillograph of a trace of data is formed, and then each trace is read and drawn as per the circulation of the starting trace and the ending trace. If the time interval becomes larger and the points are connected directly, then the curve will not be smooth but looks like broken lines, and if that happens, Bezier Curve can be drawn to improve the above. Oscillograph is composed of several small line segments, the segment vertex coordinates and the values of the seismic data trace sampling points are in one-to-one correspondence. Put the above in end-to-end connection and then the drawing of the curve can be achieved. However, 2D graph shall meet the requirements of various transformation operations, such as horizontal and vertical transformation, local area zooming. In drawing, various conditions after operation shall be given full consideration, and corresponding solutions shall be taken.

3.1. Normalization Processing

Trace gather graphics mainly adopt trace wave lines for demonstration, and the section will focus on the drawing methods and wave line zooming and transformation of trace gather wave lines. The main principles are: the CMP (common midpoint) selected by the user is used to get all the seismic record data of the CMP from the seismic record data documents, and is drawn into wave lines by traces and then is displayed on the trace Gather graphics. The whole Gather graphics are composed on a plurality of trace wave lines, and the X coordinate axis increases from left to rights by the trace number sequences, and the Y coordinate axis increases from up to down by the time axis of each trace. Each wave line is displayed in a certain width, and shall not be overlapped or interfered. Therefore, in display, the data shall receive normalization processing, namely the data shall be normalized within the range of -1--+1 in proportion as per the plus-minus amplitude to be displayed. Namely, the max plus amplitude shall be normalized as +1, and the min minus amplitude shall be normalized into -1, and +1 and -1 represent the max width of the wave line wide from left or right to the center, and the amplitude values of the max amplitude and mix amplitude are converted into the floating points between -1 and +1. The processing on the amplitude values is convenient for the conversion between numerical values and wave lines. And the display width of each wave line can be guaranteed not exceed the display range.

The values of the seismic data represent the strength of amplitudes, and the plus-minus represents the directions of vibrations; they can record the vibrations of the geophone sites at the balance position in a reasonable manner [2]. These features will not change with the transformation of the graphs, so the waveforms are the same before and after transformation. In utilization of the similarity, the waveform can be transformed into certain scope with the features of the whole waveform remained. And the following various transformation operations can be regarded as multiplying the standard waveform with a coefficient. Hence, a good solution is to transform the seismic data into the scope of $[-1, 1]$; thus not only the relative values of the data are kept, but also the plus-minus of the data represents the directions of vibration.

The specific methods for data normalization are: finding the maximum and minimum values of the data, comparing the absolute values of the two values, and taking the one of larger absolute value as the standard value. All the data values receiving mapping via the standard value [2], and its pseudo-codes are expressed as follows:

```
if fabs(minV) > fabs(maxV)
stdV = minV
Or else
stdV = maxV
mappedValue = sampleValue/stdV
```

In the codes, fabs represents “absolute value”, stdV is the mapping standard, and sample Value and mapped Value the data values in original trace and the values after mapping respectively.

3.2. Coordinate Calculation

The seismic data belong to normalized data, which mean that the data have the same distance at all directions, and that in other words, the distance of the data at the horizontal (X) and the vertical (Y) directions are the constant values respectively but the distances the X and Y directions are not the equal. Hence, the coordinates of the drawing point is related to the distances at horizontal and vertical directions. We suppose the width of each trace is trace Distance, and that the height each data point takes is sample Height. The horizontal direction is the vibration direction; therefore, X coordinates are jointly determined by the trace distance and the normalized data, whose calculation formula is illustrated in Formula (1). The trace distance in the formula determines the horizontal zooming scale of the graph, and the normalized data maintain the basic features of the waveform.

$$x = \text{m a p p e d V a l u e} \times \frac{\text{t r a c e D i s t}}{2} \quad (1)$$

The vertical direction represents the data value of the sampling points at each trace gained by certain sampling interval, it differs from the calculation of X coordinates, but is just related to the data point distance and the position of the data in the trace (namely the index). We suppose the initial index number is siy, that the ending index number is eiy, that the index of the present drawing point is j, and then the calculation formula of Y coordinates is illustrated in Formula (2).

$$y = (e i y - s i y) \times \text{s a m p l e} \quad (2)$$

4. Curve Drawing

Seismic data are ordered and scattered measured data [2], whose 2D graph is usually represented with curve. The graph display of 2D seismic data is composed of four steps, namely, model transformation (vision transformation), cutting, window view transformation, and rasterization. When the seismic data is in the final rasterization display, the pixel information of graph is in one-to-one mapping to the original seismic data, the graph after rasterization is constructed of the polygonal lines by connecting the ordered pixel data, and then the collection of these short curve segments is used to display the curve. Hence, the key to the drawing of the curve lies in the sequential drawing of the curve calculation points. Based on the above principle, in 1981, Mcallister raised the curve interpolation reconstruction algorithm of ordered points with quadratic spline curve [3]. With the study on rational curve going further and deeper, more and more researchers raised how to perform curve reconstruction on ordered curve with rational curve [4-5], and some other researchers also raised the method of reconstructing the curve of ordered points with arc spline, biarc spline or parameter polynomial spline [6].

Cubic polynomial curve is often used in daily curve table; as the control of the curve is not flexible, the above curve can not represent the curve segment passing the (interpolation) two ends and appointing to the tangent direction at the end, and the polynomial higher than cubic will increase unnecessary swinging and bring more calculation amount. See Formula (3) for the polynomial expression of cubic polynomial curve segment $Q(t)=[x(t) y(t) z(t)]$:

$$\begin{cases} x(t) = a_x t^3 + b_x t^2 + c_x t + d_x \\ y(t) = a_y t^3 + b_y t^2 + c_y t + d_y \\ z(t) = a_z t^3 + b_z t^2 + c_z t + d_z \end{cases} \quad (0 \leq t \leq 1) \quad (3)$$

Supposing:

$$T = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix}$$

The coefficient matrix of the three polynomials is defined by the following formula:

$$C = \begin{pmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \end{pmatrix}$$

Then Formula (3) can be rewritten as:

$$Q(t) = \begin{bmatrix} x(t) & y(t) & z(t) \end{bmatrix} = T * C$$

The generation of cubic polynomial curve often adopts interpolation, approximation, fitting, fairing and the like methods.

(i) Interpolation: an important method of functional approximation. Give the function $f(x)$ diverse values of n points $f(x_i)$, $i=1, 2, \dots, n$ within the interval $[a, b]$, and seek one function $g(x)$ to approximate $f(x)$ based on the table date. If $g(x)$ is required to be equal to $f(x)$ at x site, then such function approximation problem can be called interpolation problem, $g(x)$ is the interpolation function of $f(x)$, and x is the interpolation node points. At the node point $g(x)$ is equal to $f(x)$, and at other points similar $g(x)$ is used to replace $f(x)$. Usually, linear interpolation, parabola interpolation, and cubic Hermite curve are used [4].

(ii) Approximation: when there is no need to seek an interpolation function to pass all the data points, often, a function of lower times is selected to perform best approximation to the data points in certain sense. Usually, least square method, Berzier curve, and B spline are used [5].

(iii) Fairing (smoothing): making the inflection points of the curve become less and straighter and smoother. This is an interpolation method of erasing surplus inflection points. Often, tension spline function method is used [4].

(iv) Fitting: unlike the above with complete definition and mathematic expression, fitting is to make the generated curve and curve surface reach the requirements of some designs with interpolation or approximation methods in the design processes of curve or curve surfaces. Such as the data points or control sequence close to the original within the allowable scope, like B spline curve [5]. The low time polynomial interpolation by segment is used to gain the fine and smooth curve interpolation effect, and the specific steps for segment cubic polynomial interpolation are: perform classification as per the traces acquired in seismic data acquisition, and then perform grouping of the original data points of each trace as per (P0, P1, P2, P3), (P3, P4, P5, P6), (P6, P7, P8, P9), (P9, P10, P11, P12), ..., and calculate the difference quotient table with three orders are the highest order as per the grouping calculation, and finally simulate the oscillograph of each trace as per the cubic polynomial, and draw the vibration of the geophone sites as per the change of the curve. Therefore, it is easy to gain the coefficients of the cubic polynomial, but the curve is drawn by various smaller straight lines; if multi-curve shall be drawn, more curve points shall be provided, and then the final curve graph can be drawn by approximation of the straight lines between neighboring data points. Thus, calculation

amount will be increased, but curve of higher precision will be gained. Therefore, in drawing the oscillograph of 2D seismic data, the paper adopts segment cubic time Bezier curve. Besides, when there is few original data or the original data are amplified, the partition and fitting methods of Bezier curve are used to solve the weak smoothness problem.

If a point is connected to a point directly, the curve will not look smooth but look like broken lines, we can improve the curve by drawing Bezier Curve. Bezier Curve is a parametric polynomial curve based on approximation. Curve fitting is actually a kind of interpolation. When there is few data points, a series of new vertexes can be generated on basis of the known points, and then the curve is generated with the original data points, which can make the cure smooth and natural.

4.1. The Generation of Bezier Curve

Bezier Curve is a kind of parametric polynomial curve based on approximation, and curve approximation is actually a kind of interpolation. The Graphics category of GDI+ provides the function of drawing Bezier Curve, namely Graphics::DrawBeziers(Pen* pen, PointF*points, INT count), and the function has three parameters: the pointer pointing to the paint brush, the array for storing the control point coordinate and the quantities of coordinates. We just need to group the seismic data, and the coordinates corresponding to the group of the data points are stored in the array as the control points; after a Graphics object and a Pen object are instantiated, the function is called to pass parameters, and an internally closed algorithm is used to draw Bezier Curve of itself. As is known from the definition of Bezier Curve, Bezier curve is a kind of parameter polynomial curve, which is generated by the polynomial curve provided in Formula (3). But in this manner, for the high times curve, the calculation amount of the algorithm is heavy, but the De Casteljaun algorithm based on Bernstein primary function recurrence will be more efficient.

De Casteljaun Algorithm: give the space $n+1$ points P_i ($i=0,1,\dots,n$) and parameter t , and then the recursion formula shall be:

$$P_i^r(t) = (1-t)P_i^{r-1}(t) + tP_{i+1}^{r-1}(t) \quad \begin{cases} r=1, \dots, n \\ i=0, \dots, n-r \end{cases} \quad \text{Wherein, } P_i^0(t) = P_i, \quad P_0^n(t) \text{ is the point}$$

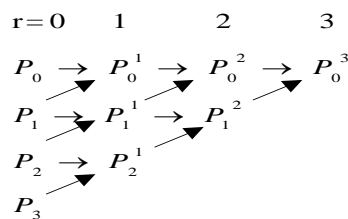
whose parameter is t on Bezier curve.

The calculation process and geometrical significance of De Casteljaun algorithm are illustrated in Figure 1 (a), (b), and the calculation steps are as follows:

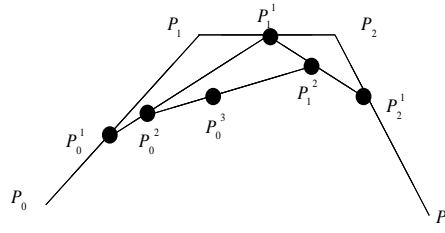
(i) Via the parameter value t , determine the break point at each scope of the polygon, and thus make the segment ratio after break t : $(1-t)$, namely $P_i^1 = (1-t)P_i + tP_{i+1}$, $i=0, 1, \dots, n-1$, and thus a new polygon with $(n-1)$ sides is formed.

(ii) Perform partition on the new polygon with the same methods, then the break points P_i^2 ($i=0, 1, \dots, n-2$) are gained, and another new polygon is formed.

(iii) After $n-1$ times of partition, two points P_0^{n-1} and P_1^{n-1} are gained, and P_0^n is gained after another partition, and the point is the t data point with a parameter of t on Bezier curve.



(a) Cubic Time Bezier Curve Calculation Process



(b) Cubic Time Bezier Curve Geometric Construction Figure

Figure 1. De Casteljau Algorithm Design

The trace date display in the paper adopts Qt language to perform programming, The code of De Casteljau is as follows:

```

int deCasteljau(CDC *pDC, CArray<float, float>& flArrayx, CArray<float, float>&
flArrayy)
{
if(flArrayx.GetSize() != flArrayy.GetSize())
return 0;
float *pflX, *pflY;
float flTempx, flTempy, flU;
int i, n, j;
n = flArrayx.GetSize();
if(n < 2) return 0;
pflX = new float[n];
pflY = new float[n];
flTempx = flArrayx.GetAt(0);
flTempy = flArrayy.GetAt(0);
for(i = 0; i < n; i++) {
pflX[i] = flArrayx.GetAt(i);
pflY[i] = flArrayy.GetAt(i);
}
for(flU = 0; flU <= 1; flU += 0.05/n)
{
for(i = 1; i < n; i++)
{
for(j = 0; j < n - i; j++)
{
pflX[j] = (1 - flU) * pflX[j] + flU * pflX[j + 1];
pflY[j] = (1 - flU) * pflY[j] + flU * pflY[j + 1];
}
}
pDC->MoveTo(flTempx, flTempy);
pDC->LineTo(pflX[0], pflY[0]);
flTempx = pflX[0];
flTempy = pflY[0];
}
delete[] pflX;
delete[] pflY;
return 1;
}
    
```

Algorithm 2 Bezier Curve Drawing Algorithm


```

Start:
Void drawBezier(Qpoint P [],int n,int m,QpainterPath*pth)
// P is the control point, n is the number of the control points, and m is the equal
fraction on the interval [0, 1].
{
    Double u=1.0/m;
    Qpoint *q=new QPoint[m+1];//the data point on the curve, M+1in total
    q[0]=p[0];
    //Call DeCastel jau Algorithm to calculate the data point on the curve
    for(int i=1;i<m;i++)
        q[i]=DeCastel jau(p,n-1,u*i);
    For(i=0;i<n-1;i++/*show the featured polygon */
    {
        Pth->moveTo(p[i].x(),p[i].y());
        Pth->lineTo(p[i+1].x(),p[i+1].y());
    }
    For(i=0;i<m;i++)/*use polygonal line to express Bezier Curve */
    {
        Pth->moveTo(q[i].x(),q[i].y());
        Pth->lineTo(q[i+1].x(),q[i+1].y());
    }
    Delete [ ]q;
}
end

```

4.2. The Segmentation of Bezier Curve

The Bezier curve P is divided into Section C and Section Q from a certain part in the middle, and curve Sections C and Q are also curve, and their control vertexes are generated by De Casteljau Algorithm. P_0^3 Then the whole Bezier curve will be divided into two unequal subset curve segments, the first half of the Bezier curve is mainly used to control the parameter interval vertexes $P_0^3 P_0^1 P_0^2 P_0^3$, and latter half of the Bezier curve is mainly used to control the parameter interval vertexes $P_0^3 P_1^2 P_2^1 P_3$, and their parameter intervals are $[0, t]$ and $[t, 1]$. In other words, by giving arbitrary parameter value $\mu \in (0,1)$, the point $P(\mu)$ on Bezier curve can divide the curve into two sub curve segments $[0, \mu]$ and $[\mu, 1]$, which are marked as C and Q respectively. De Casteljau algorithm can get the horizontal side vertex set $\{P_o^i = 0, 1, \dots, n\}$ of the triangle by recurrence to define the sub curve, the control vertexes $\{P_i^{n-i} = 0, 1, \dots, n\}$ of Curve Q form the hypotenuse of the triangle, and the vertical sides of the triangle are formed by the control vertex of the original curve.

C. The Reverse Engineering of Bezier Curve Vertex

Bezier Curve is a parametric polynomial curve based on approximation, multilateral polylines. As per the only definition of each peak on a group of polylines, in the peaks of the polylines, only the first point and the last point are on the curve, and the rest peaks are used to define the orders of the curve's parameter equation and the shape of the curve. In given $P_i (i=0,1,2,\dots,n)$, there are $n+1$ control points, then the n power Bezier Curve is:

$$Q(t) = \sum_{i=0}^n P_i B_i, t(t) \quad t \in [0,1]$$

In the formula:

P_i represents the feature polygon that forms the curve.

$B_{i,t}(t)$ is the harmonic function of each site on the curve.

$B_{i,t}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} = C_n^i t^i (1-t)^{n-i}$ $i=0,1,\dots,n$. When $n=3$, there are 4 control

points, and the interpolation formula is as follows:

$$\sum_{n=1}^{\infty} P_i P_{ic_3}(t) = (1-t)^3 P_0 + 3t(1-t)^2 + 3t^2(1-t)p_2 + t^3 p_3 \quad 0 \leq t \leq 1$$

The matrix form is:

$$Q(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ 1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ 1 & 3 & -3 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} Q(t)$$

From the above, $P_0=Q(0), p_3=Q(1), p_1, P_2$ can all be gained by solving, the control point of Bezier Curve can be gained as per the above inverse computation, Bezier Curve can be drawn by Qt with the help of the control point, and the codes are as follows:

```
QPainter Path Path;
Path.moveTo(P0);
Path.cubicTo(p1,p2,p3);
QPainter Painter(this);
Painter.draw Paht(Path);
```

For given $n+1$ points $Q_i(0,1,\dots,n)$, if a Bezier curve shall be constructed to pass these points, then the control vertexes $P_i(0,1,\dots,n)$ of Bezier Curve shall be gained. Usually, the parameters $t=i/n$ corresponding to Q_i shall be used for the reverse engineering of P_i . As per the definition of Bezier Curve, $n+1$ linear equation groups can be gained:

$$\begin{cases} Q_0 = P_0 \\ \vdots \\ Q_i = P_0 C_n^0 (1-i/n)^n + P_1 C_n^1 (i/n)(1-i/n)^{n-1} + \dots + P_n C_n^n (i/n)^n, i \in [1, n-1] \\ \vdots \\ Q_n = P_n \end{cases}$$

Thus, the control vertexes $P_i(i=0,1,\dots,n)$ of Bezier Curve can be gained by the equation groups, and the n times Bezier Curve of Q_i can be generated by the reverse engineering of the control vertex P_i .

5. Test Results

In combination with the above oscillograph and Bezier curve fitting, the paper, with the aid of the measured data, draws the data result figure as per the actual data, and the figure

is illustrated in Figure 4. As can be learned from Figure 2, as each row is in dense arrangement, the data points requiring acquisition are quite many, and the data points are of high complexity, so no curve fitting shall be required, and the comprehensive features of the reflection surface can be expressed; while in Figure 3, there are few traces and data points to draw, so Bezier curve fitting will make the waveform more smooth and in more natural connection, and thus expression particle vibration can be expressed better.

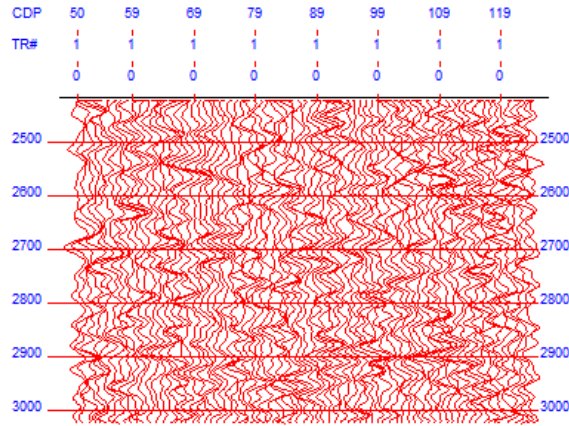


Figure 2. Bezier Curve is Unfit for the Scene with Large Data Amount

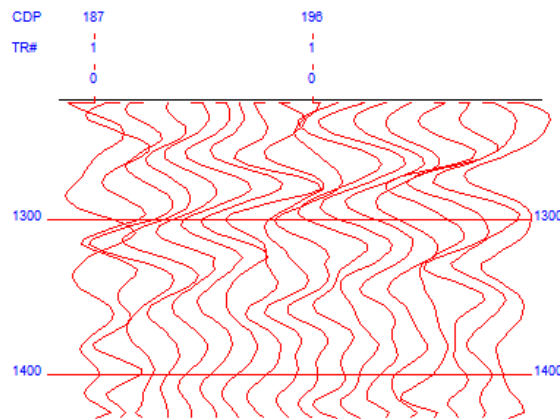


Figure 3. When there is Few Data Points; Bezier Curve Fitting Shall be Utilize

6. Summary

Based on the analysis of SEG-Y data format, the paper expounds the drawing of seismic trace data oscillograph in details, based on the features of the actual trace data, gives the algorithm of Bezier Curve fitting when the trace sampling data points are few or there are few sampling points to be drawn due to the zooming of graph, and gain the expected results in actual test results. The waveform becomes smoother and the connection more natural with the use of Bezier Curve and the particle vibration is expressed better, which is helpful for the engineer to perform analysis and interpretation on the explored seismic data.

Acknowledgments

Project number: KJ091306. Corresponding author: Wenqin Li

References

- [1] Z. Wan, W. Lei and T. Jingyong, "Visualization of SEG Y Seismic Data Based on Qt in Window", *Computer and Digital Engineering*, vol. 36, (2008), pp. 10-12.
- [2] Yilmaz O., "Seismic Data Processing", *Society of Exploration Geophysicists*, vol. 21, no. 3, (1991), pp. 42-49.
- [3] McAllister D. F. and Roulier J. A., "An algorithm for computing a shape preserving oscillatory quadratic spline", *ACM Transactions on Mathematical Software*, vol. 16, no. 4, (1991), pp. 331-347.
- [4] Sarfraz M., "A rational cubic spline for the visualization of monotonic data", *Computers & Graphics*, vl. 6, no. 24, (2000), pp. 509-516.
- [5] Wang Q., Ward R. and Sun H. J., "Isophote estimation by cubic—spline interpolation in Proceedings of Image Processing", Munich Germany, (2002), 24-18.
- [6] Sohel F. A., Karmakar G. C. and Dooley L. S., "A dynamic Bezier curve model", *Image Processing*, vol. 15, no. 2, (2005), pp. 11-14.
- [7] T. Rongxi, "Modern Graphics Technology", Jinan: Shandong Science and Technology Press, (2002), pp. 326-355.
- [8] Tomas A. M. and Eric H., "Real Time Rendering", Beijing: Peking University Press, 2nd Edition, (2004), pp. 14-39.
- [9] H. Gang, L. Fang, L. Hai and S. Jingguang, "Multi-attribute Trilateral Filtering for Seismic Data", *Journal of Computer-Aided Design & Computer Graphics*, (2010).
- [10] G. Hua and H. Lin, "Tensor Based Feature Enhancement for 3-D Seismic Data Visualization", In Proceedings of 2009 Workshop on Visualization Applications Collocated with IEEE Pacific Vis, (2009).
- [11] G. Hua, H. Lin and J. G. Sun, "A New Quantization Method for 3-D Seismic Visualization", In Proceedings of 11th IEEE International Conference on Computer-Aided Design and Computer Graphics, (2009), pp. 457-462.
- [12] G. Hua, B. Tang, Z. Zhou, J. Shuai and H. Lin, "Interval Volume Rendering using Ray Opacity Modulation", Submitted to Eurographics/IEEE-VGTC Symposium on Visualization, (2011).
- [13] Z. Zhou, J. Shuai, G. Hua, B. Tang and H. Lin, "Feature-preserving Quantization for 3D Seismic Visualization", Submitted to Eurographics IEEE-VGTC Symposium on Visualization, (2011).
- [14] C. Jia, T. C. Keung and W. Jue, "Noise brush: interactive high quality image-noise separation", *ACM Trans. Graph.*, vol. 28, no. 5, (2009), pp. 146:1 — 146:10.
- [15] Buades A., B. Coll and J. Morel, "Nonlocal Image and Movie Denoising", *International Journal of Computer Vision*, (2008), pp. 123-139.
- [16] Cipriano G., G. N. Phillips Jr. and M. Gleicher, "Multi-Scale Surface Descriptors", *IEEE Transactions on Visualization and Computer Graphics*, (2009), pp. 1201-1208.
- [17] Hosssain Z. and T. Moller, "Edge Aware Anisotropic Diffusion for 3D Scalar Data", *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, (2010), pp. 1376-1385.
- [18] Carr H., J. Snoeyink and M. Van de Panne, "Flexible iso surfaces: Simplifying and displaying scalar topology using the contour tree", *Computer. Geom. Theory Appl*, (2010), pp. 42-58.
- [19] Kadlec B. J., H. M. Tufo and G. A. Dorn, "Knowledge-Assisted Visualization and Segmentation of Geologic Features", *IEEE Computer. Graph. Appl*, (2010), pp. 30-39.
- [20] Ament M., D. Weiskopf and H. Carr, "Direct Interval Volume Visualization", *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, (2010), pp. 1505-1514.
- [21] A. M. Al-Canaan and A. Khoumsi, "Multimedia Web Services Performance: Analysis and Quantification of Binary Data Compression", *Journal of Multimedia*, (2011), pp. 447-457.
- [22] Z. H. Wang, C. C. Chang, M. C. Li and T. C. Lu, "A Simple Image Encoding Method with Data Lossless Information Hiding", *Journal of Multimedia*, (2011), pp. 400-407.
- [23] X. Wang, G. Yao and Z. Zhang, "An Encryption Scheme with Hidden Keyword Search for Outsourced Database", *Journal of Multimedia*, (2011), pp. 1697-1704
- [24] M. Li, Z. Liu, J. Li and C. Jia, "Format-Preserving Encryption for Character Data", *Journal of Multimedia*, (2012), pp. 1239-1244
- [25] L. Wenqin and W. Daqing, "Visualization System of Massive 2D Seismic Data", *Journal of Software*, (2012), pp. 1625-1632.

Authors

Wenqin Li was born at Quxian County, Sichuan Province, China on October 1, 1976. She got her bachelor's degree of engineering in the field of education of computer science in College of Computer Science, Southwest Normal University, Chongqing, China in July 1999. She also got her master's degree of engineering in applied technology of computer science in College of Computer Science, Chongqing University, Chongqing, China in June 2008.

She has worked as a teacher in Yangtze Normal University since 1999 and later in 2004 she was promoted as a lecturer. She is mainly engaged in teaching of computer professional course in the computer school of her university. The major courses she teaches include computer composition principle, software engineering, computer aided design, *etc.* What interest her are data mining and education informationization.

She has some articles published, including The Analysis And Application of Workflow Technology and Petri Net for Modeling in Educational Management Software. Chongqing, China: Journal of Chongqing Technology and Business University (Natural Science Edition), 2011, Vol. 02, Visualization System of Massive 2D Seismic Data in Journal of Software, 2012, Software Project Management and Risk Analysis, in Automation & Instrumentation, 2009, Vol.06, and Constructing Experimental Platform of Automation Engineering, in Automation & Instrumentation, 2010, Vol.01.

