Journal of Computations & Modelling, vol.3, no.4, 2013, 311-329
ISSN: 1792-7625 (print), 1792-8850 (online)
Scienpress Ltd, 2013

Graphic translation algorithm research and the realization of man-machine interaction in the semi-logarithmic coordinate system

Liu Bo-tao[1], Wang Xin-Hai[1], Zhang Fu-Xiang[2],

Huang Jia-qiang[3] and He Xiu-Ling[1]

Abstract

The research of design and implementation of graphics translation function has a practical value for the software development. Especially in the development of well test interpretation software, formation parameters need to be calculated by parallelly moving the chart curve in the log-log coordinate system. Therefore, in the log-log coordinate system, the research of design and implementation of graphics translation function becomes a necessary part for the development of such similar software. In this paper, the author has systematically expounded the

[1] Key Laboratory of Oil and Gas Resources and Exploration Technology, Ministry of Education, Yangtze University, Jingzhou, Hubei, 434023, China.
Email: liubotao920@163.com.
[2] China National Petroleum Corporation Tarim Oilfield, Korla, Xinjiang, 841000, China.
[3] Langfang branch, Research Institute of Petroleum Exploration & Development, PetroChina, Langfang, Hebei, 065007, China.

Article Info: *Received* : September 23, 2013. *Revised* : November 20, 2013.
Published online : December 1, 2013.

graphics coordinate transformation formula under three different kind of coordinates: ordinary rectangular, log-log and semilog coordinate system, has showed the design ideas and relevant algorithms of the graphics translation function, and has given the implementation process of user interaction, relevant flowcharts and key codes, subsequently has taken the well test interpretation software as an example, which is developed under the VS2010 environment, and has carried out a detailed test through the way mentioned above. And the test results show that the software, designed and implemented by the ideas, runs correctly and reliably.

Keywords: Semi-logarithmic Coordinate System; Translation; Coordinate Transformation; Artificial Interaction; VS2010.

1 Introduction

In the course of development of well testing interpretation software, a log-log chart curve usually needs to be drawn [1-3] and to be moved to calculate the formation parameters [3]. Therefore, in the semi-logarithmic coordinate system, graphics translation function is necessary for development of such well test interpretation software.

According to the investigation, the literature [4] used 2D graphics transformation theory, gave the coordinate transformation formula of graphics translation, and gave the program realization. The literature [5] introduced a method of multifunctional drawing window, and realized a drawing window of graphics translation function. The literature [6] used the method of graphics combined transformation and matrix inversion to infer the coordinate transformation between world coordinate and partial coordinate, and the basic translation transformation matrix is given. The literature [7] presented a method, which can do graphics translation real-timely according to man-machine interaction selectively, and gave the program realization. The literature [8] offered parallel moving process and transformation formula and calculated with examples. The literature [9] gave a program implementation of graphics translation in VB. The literature [10] proposed a general method, which can search the corresponding transformation matrix among the different 3D coordinate system, and gave the transformation matrix of graphics translation among three kinds of coordinate system in 3D rectangular coordinate system. The literature [11] presented the transformation matrix of graphics translation among three kinds of coordinate system under the 2D rectangular coordinate system. The literature [12] discussed translation method of different distribution curve models, and established translation formula of composite hyperbolic tangent function model, and presented the translation formulas of several commonly used distribution curve model. The literature [13] presented the VC++ implementation of smooth translation curve. The literature [14] derived the general form of transformation matrix of several kinds of graphics geometric transformation, including translation transformation. The literature [15] achieved the real-time translation of graph in the rectangular coordinate system by changing the way of the screen coordinate. The literature [16] used the transformation matrix to realize the coordinate transformation of VRML modeling from local coordinate system to world coordinate system and gave the point coordinate and their translation transformation formulas. The literature [17] studied the principle of vector graphics translation transformation in Windows GDI+ from the perspective of computer graphics, and gave some example codes.

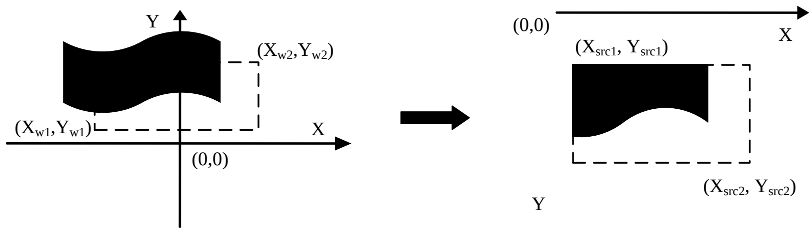
As can be seen from the above, almost all the literatures discussed the translation in 2D or 3D rectangular coordinate system without considering the translation under the log-log or the semilog coordinate system.

The author will discuss the ordinary rectangular, the log-log and the semilog coordinate system, expound the principle and formula of coordinate transformation, discuss the design of graphics translation function, deduce the algorithm design process and result, and present its VS2010 implementation and test results.

2 The Algorithm Research

2.1 The Algorithm Research of Coordinate Transformation

Graphics drawing is based on key points, thus the study of graphics coordinate transformation should begin on the basis of the coordinate transformation of the point coordinate. In order to make it clear that how to translation point under ordinary rectangular, semilog and log-log coordinate system, the chief problem is to figure out how to map the key point of graphics under the world coordinate system into the window in the screen.



a) The Interest Plot Area under World Coordinate System b) The Display Graphics by Mapped to Screen Window

Figure 1: The Demo of the Mapping Process

There are many kinds of world coordinate system when divided in detail, the author takes the 2D plane rectangular coordinates system for an example, as is shown in Figure 1. The specific way of mapping can also have a lot of kinds, Figure 1 uses "monolithic moving" to mapping, the advantage of doing so is to maintain the graphics "linear consistency" before and after mapping. And its mapping rules are defined as follows: 1) In Figure (1-a), there are two perpendicular intersect real-number lines extending to infinity. The intersection point of two lines is called the origin (0, 0), and coordinate of any point on graphics is made up by two numerical values (X_w, Y_w), and X_w, Y_w are likely to be negative, may also be a real number; 2) And in Figure (1-b), there is the coordinate system on a computer screen, its origin is the upper left corner of the drawing window's client area(by default), the X axis is extending to the right direction, Y axis is down, its value of all coordinates are positive. Coordinate axis is not a real number line, but integer line, and coordinate values are all integers, rather than real[18]; 3) Assuming that the drawing canvas on the position of the drawing window's client area is as shown in the dashed frame of Figure (2-b), the upper left corner point coordinate is (X_{src1}, Y_{src1}), and the lower right corner point coordinate is (X_{src2}, Y_{src2}); 4) The point (X_{w1}, Y_{w1}), which is the minimum X, Y coordinate values in the world coordinate, is mapped to the point (X_{src1}, Y_{src2}) of the canvas screen coordinate; 5) The point (X_{w2}, Y_{w2}), which is the maximum X, Y coordinate values in the world coordinate, is mapped to the point (X_{src2}, Y_{src1}) of the canvas screen coordinate; 6) Assuming that any other point (X_w, Y_w) in the world coordinate is mapped into the point (X_{src}, Y_{src}) in screen coordinate, as shown in Figure 2; 7) In the process of mapping, two graphics before and after translation all meet the graphics "linear consistency", when the user move the graphics parallelly in the screen coordinate space.

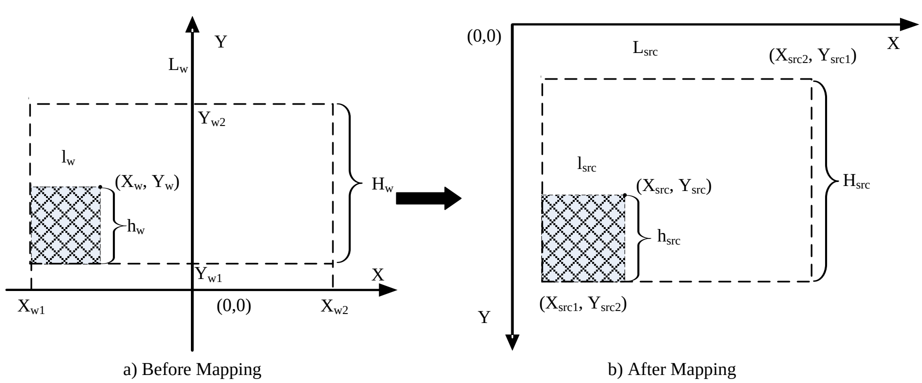


Figure 2: The Demo of the Mapping Algorithm

Due to the mapping of linear relationship: Meaning that the projection, generated by the mapping window size on the X, Y axis, meets the linear relationship before and after mapping, that is to say, here is the following relationship:

$$\frac{l_w}{L_w} = \frac{l_{src}}{L_{src}} \tag{1}$$

$$\frac{h_w}{H_w} = \frac{h_{src}}{H_{src}} \tag{2}$$

And because the Figure (3-b) is screen coordinate, ordinary rectangular coordinate system, thus there are:

$$\frac{l_{src}}{L_{src}} = \frac{X_{src} - X_{src1}}{X_{src2} - X_{src1}} \tag{3}$$

$$\frac{h_{src}}{H_{src}} = \frac{Y_{src2} - Y_{src}}{Y_{src2} - Y_{src1}} \tag{4}$$

2.1.1 The Graphics Transformation Algorithms under Rectangular Coordinate System

If the coordinate in Figure (3-a) is an ordinary rectangular coordinate system, then:

$$l_w = \mathrm{X}_w - X_{w1} \tag{5}$$

$$L_w = \mathrm{X}_{w2} - X_{w1} \tag{6}$$

$$h_w = \mathrm{Y}_w - Y_{w1} \tag{7}$$

$$H_w = \mathrm{Y}_{w2} - Y_{w1} \tag{8}$$

Put the formula (3), formula (5) and formula (6) into the formula (1):

$$X_{src} = X_{src1} + \frac{\mathrm{X}_w - X_{w1}}{\mathrm{X}_{w2} - X_{w1}} (X_{src2} - X_{src1}) \tag{9}$$

Similarly, put the formula (4), formula (7) and formula (8) into the formula (2):

$$Y_{src} = Y_{src2} - \frac{Y_w - Y_{w1}}{Y_{w2} - Y_{w1}}(Y_{src2} - Y_{src1}) \tag{10}$$

Formula (9) and formula (10) are transformation formulas between the world coordinate and the screen coordinate under the rectangular coordinate system.

2.1.2 The Graphics Transformation Algorithms under Log-Log Coordinate System

If the coordinate in Figure (3-a) is a log-log coordinate system, there are:

$$l_w = \log_{10} X_w - \log_{10} X_{w1} \tag{11}$$

$$L_w = \log_{10} X_{w2} - \log_{10} X_{w1} \tag{12}$$

$$h_w = \log_{10} Y_w - \log_{10} Y_{w1} \tag{13}$$

$$H_w = \log_{10} Y_{w2} - \log_{10} Y_{w1} \tag{14}$$

Put the formula (3), formula (11) and formula (12) into the formula (1):

$$X_{src} = X_{src1} + \frac{\log_{10}(X_w / X_{w1})}{\log_{10}(X_{w2} / X_{w1})}(X_{src2} - X_{src1}) \tag{15}$$

Similarly, put the formula (4), formula (13) and formula (14) into the formula (2):

$$Y_{src} = Y_{src2} - \frac{\log_{10}(Y_w / Y_{w1})}{\log_{10}(Y_{w2} / Y_{w1})}(Y_{src2} - Y_{src1}) \tag{16}$$

Formula (15) and formula (16) are transformation formulas between the world coordinate and the screen coordinate under the log-log coordinate system.

2.1.3 The Graphics Transformation Algorithms under Semilog Coordinate System

In semilog coordinate system, X axis acts as the logarithmic coordinate, and Y axis acts as common coordinate, which are similar to the well test interpretation software in the literatures [1-3], thus when map on the X axis, the process that turn the world coordinate value to screen coordinate value can be calculated by formula (15), and formula (10) will be used when map on the Y axis.

Similarly, if X axis is a normal coordinate and Y axis is the logarithm coordinate, then use formula (9) and formula (16) respectively.

Thus, semilog coordinate system can be derived analogically by the ordinary rectangular and log-log coordinate system. Therefore, semilog graphics transformation will not be discussed in the following part.

2.2 The Algorithm Research of Graphics Translation

2.2.1 Graphics Translation Function Design

There are three methods for the design of curve translation function: 1) translation through mouse interaction: through the course of moving the mouse, the mouse changes style when meets the curves that need translation. Then press the mouse to drag the curve at this time; don't release the left button of the mouse until the curve being moved to the specified location. Finally, the curve translation is completed; 2)translation through the keyboard interaction: after the curve need to translation being selected by the user, if click on one of the four Up/Down/Left/Right navigation keys, the curve will be Up/Down/Left/Right adjusted drastically, If the size need to be adjusted slightly, just press one navigation key and Shift key together, if you want to adjust the size much more slightly, you can press one navigation key and Control key together. 3) According to the given increment of X axis and Y axis, the software realizes automatic translation.

No matter which scheme the translation is, all is based on translation algorithm.

2.2.2 Graphics Translation Algorithm Research

As is shown in Figure 3, (3-a) is the world coordinate system space, however, Figure (3-b) is the screen coordinate space. Assuming that the coordinate value of any i-th point of the curve in the Figure (3-a) is (x_{wi1}, y_{wi1}), and after this point being mapped to screen canvas, the coordinate value is (x_{srci1}, y_{srci1}). Assuming that the projection of curve translational increment (can be positive or negative) on X axis and Y axis is Δx and Δy respectively, and the coordinate values of the point become (x_{wi2}, y_{wi2}) and (x_{srci2}, y_{srci2}) in the two space coordinate after the translation.

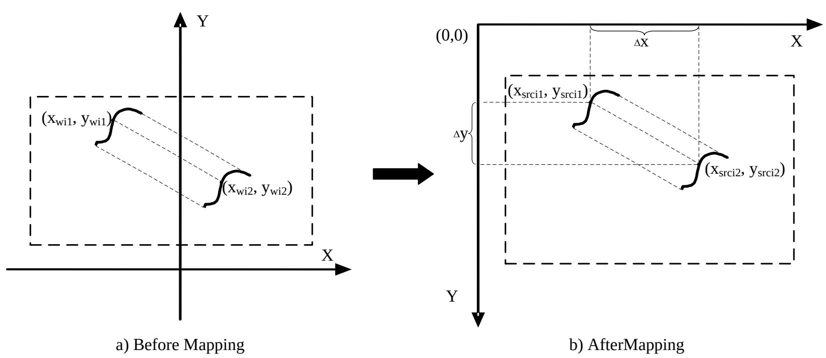


Figure 3: The Demo of the Graphics Parallel Moving Algorithm

Because the Figure (3-b) is drawing area which is the ordinary rectangular coordinate system, so there is:

$$x_{srci2} = x_{srci1} + \Delta x \tag{17}$$

$$y_{srci2} = y_{srci1} + \Delta y \tag{18}$$

Therefore, curve translation algorithm can be described as follows in Figure 4.

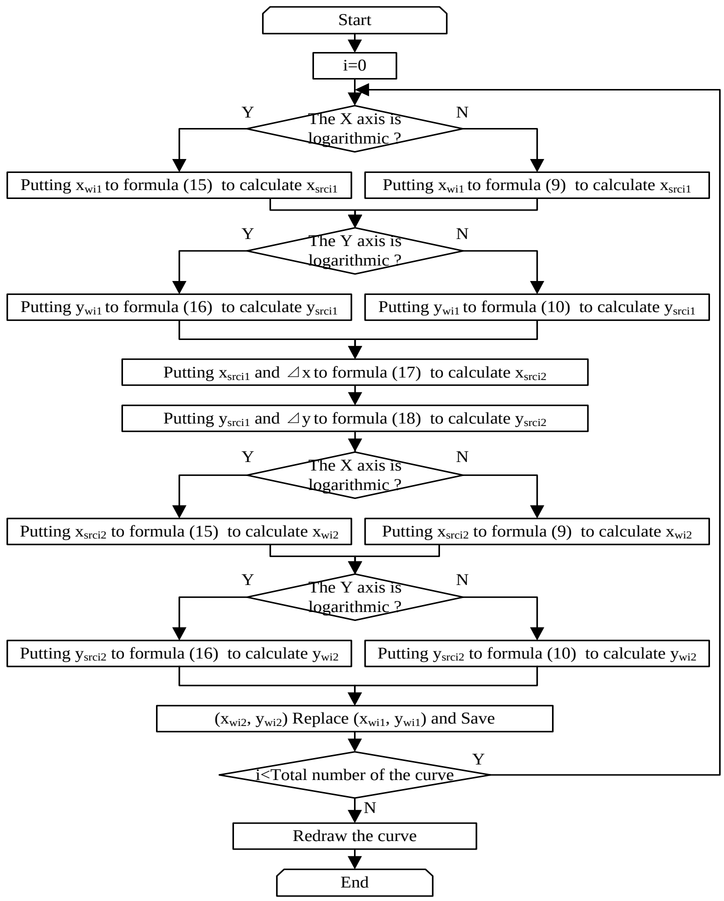


Figure 4: The Flow Chart of the Curve Translation Algorithm

2.3 Translation Function Program Implementation

2.3.1 The Realization of Mouse Interaction Translation

Assuming that the canvas window class is CCanvasView (the same below), implementation is as follows:

1) Add some member variables of class CCanvasView and initialize them as is shown in Table 1.

Table1: The Member Variables and Related Information Needed to Add

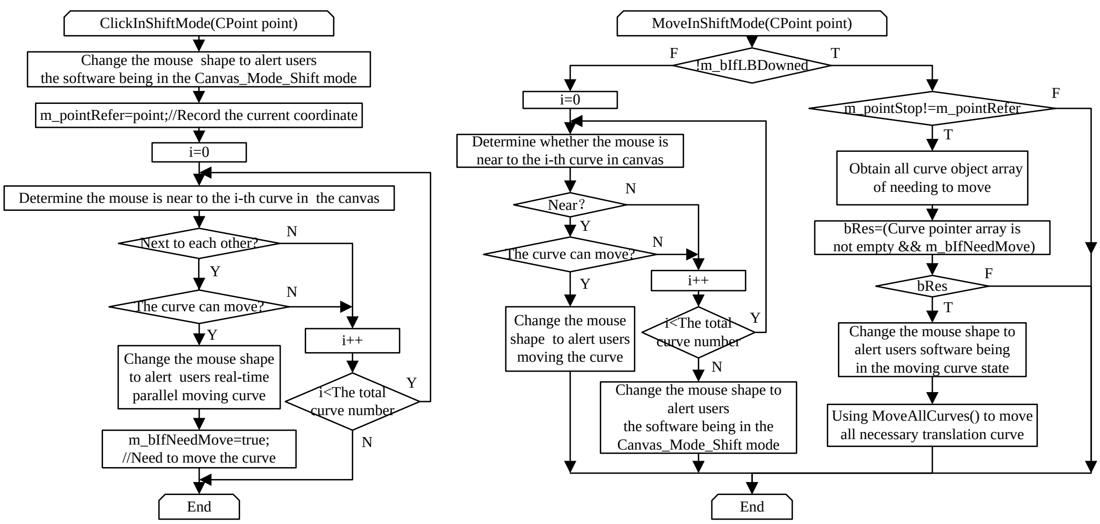
Variable Name	Variable Types	Initial Value	Definition Purpose
m_bIfLBDowned	bool	false	Whether the left mouse button has been pressed or not, is used to distinguish between the mouse drag and translation.
m_bIfNeedMove	bool	false	Indicate that whether there is a curve need to be translated or not.
m_pointRefer	CPoint	CPoint(-1,-1)	The reference point of translation.
m_pointStop	CPoint	CPoint(0,0)	Record the corresponding screen coordinate at the end of mouse drag.

2) Add the following codes to the funtion CCanvasView::OnLButtonDown():

m_bIfLBDowned=true; //Press the left key of mouse

if(Canvas_Mode_Shift==m_enumMode) { ClickInShiftMode(point); }

m_enumMode is a member variable of the CCanvasView class, identifying the current mode of the canvas. And the Canvas_Mode_Shift represents curve translation mode. The realization process of function ClickInShiftMode () is shown in Figure (5-a).



a) b)

Figure 5: Realization Process of ClickInShiftMode() and MoveInShiftMode()

3) Add the following codes to the funtion CCanvasView::OnMouseMove():

```
if(Canvas_Mode_Shift==m_enumMode){
m_pointStop= point; //Record current screen coordinate of the mouse
MoveInShiftMode(point);
m_pointRefer= point;//Always use current screen coordinate of the mouse as a reference point for translation
}
```

Among them, the realization process of the function MoveInShiftMode() is shown in Figure (5-b), it calls the function MoveAllCurves() to translate all curves needed to be translated. The realization process of the function is as shown in Figure (6-a), and has three parameters in total: The first parameter is the CTypedPtrArray<CPtrArray, pointer set of CDynamicCurve*>& type, pointing to all curves to be translated; the second and the third parameters is CPoint type, recording the initial and terminate screen coordinate of the curve. It calls the MoveOneCurve() function for a concrete translation. The realization process of the function is as shown in Figure (6-b), and has three parameters: the first parameter is the object pointer of the translated curve. The other two parameters point the projection of the displacement increment of curve on the X axis and Y axis respectively.

4) Add the following codes to the funtion CCanvasView::OnLButtonUp():

```
m_bIfLBDowned=false;//The left mouse button is pop-up
if(Canvas_Mode_Shift==m_enumMode){//End translation process
m_pointStop= point;
if(m_pointStop!=m_DragStartPoint){//relative to the initial reference point，the displacement occurred
UpInShiftMode();
}}
```

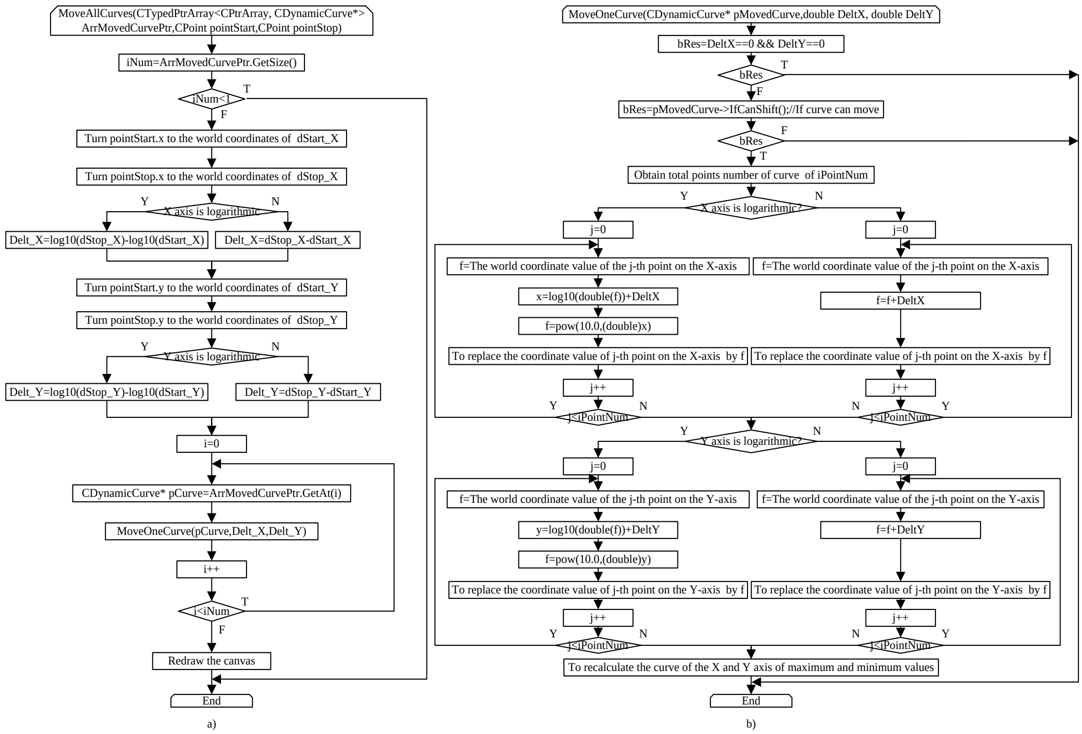


Figure 6: The Realization Process of MoveAllCurves() and MoveOneCurve()

Among them, the realization process of the function UpInShiftMode() is as shown in Figure 7.

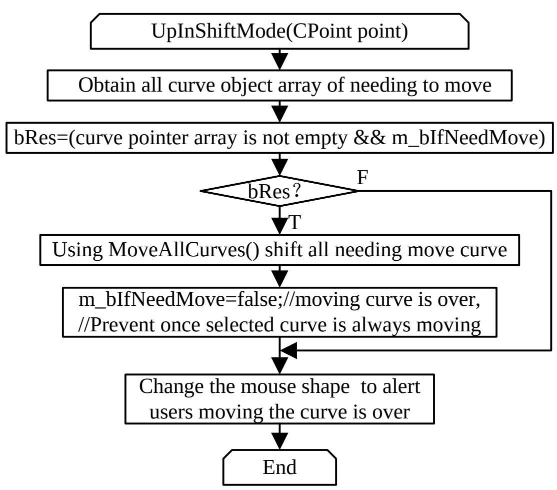


Figure 7: The Realization Process of UpInShiftMode()

2.3.2 The Realization of Keyboard Interactive Translation

When considering keyboard to achieve interaction, people usually choose to use hotkeys, using API function RegisterHotKey() to register the hotkey, UnregisterHotKey() to uninstall the hotkey and OnHotKey() in response to the hotkey function to carry out specific translation processing. But the weakness of such implementation is: Once the keys are used in other software, it will produce conflict, causing "hotkey failure" in other software. Therefore, the author directly captures the user's keyboard events, analyze and process keyboard interaction in keyboard events. Implementation method is to overload function CCanvasView::PreTranslateMessage(MSG* pMsg), add the following code:

```
if(WM_KEYDOWN == pMsg->message) {// When pressing the keyboard
MoveCurveByKeyboard();}
```

Among them, call the function MoveCurveByKeyboard() to realize the translation of the curve, and the implementation process is as shown in Figure 8.

2.3.3 The Realization of Automatic Translation in the Software

From all above we can know that it is very simple to realize automatic translation: 1) Be clear to the pointer set of curves need to be translated; 2) Call the function MoveAllCurves() to realize translation of all curves; 3) Redraw the canvas.

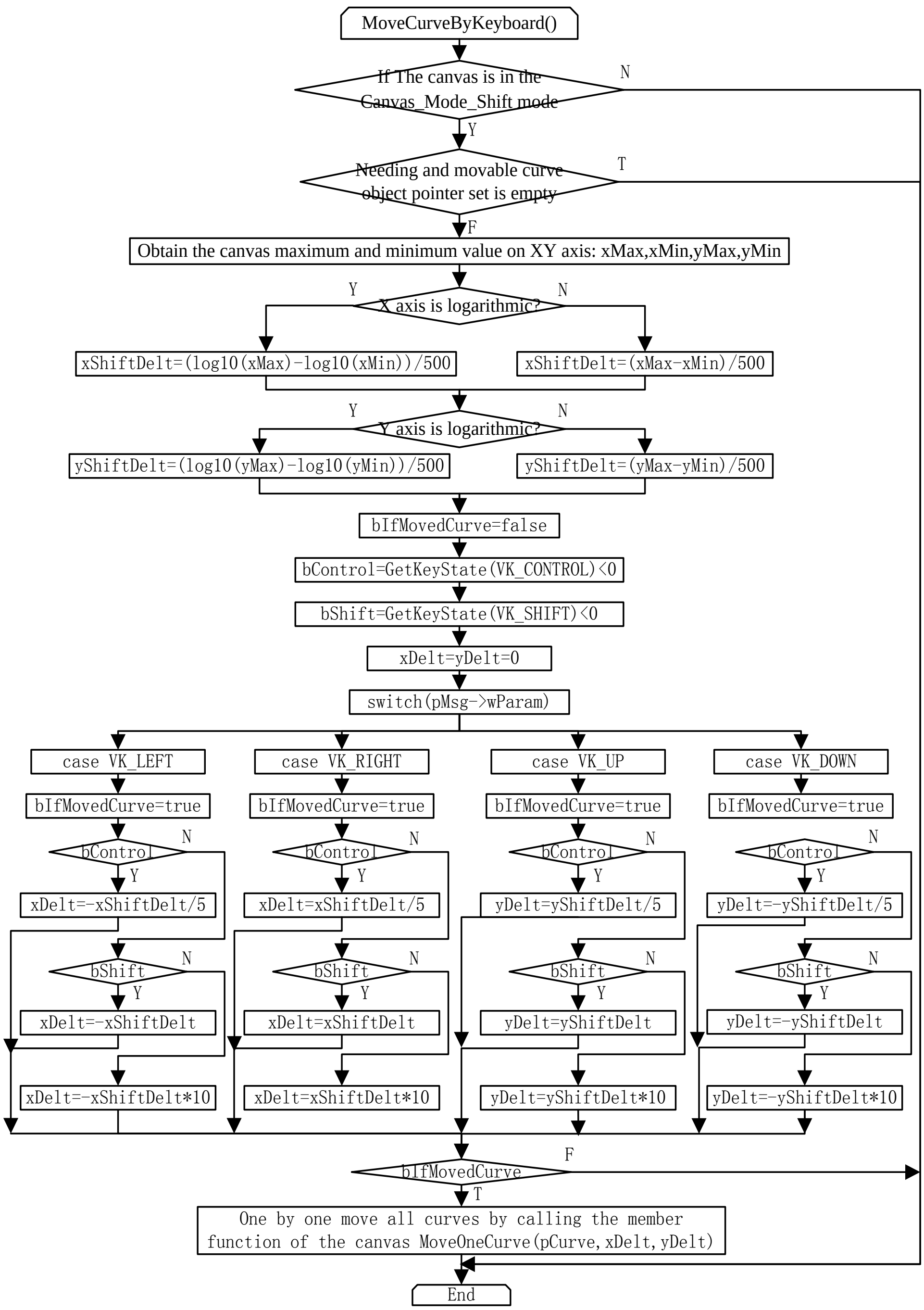


Figure 8: the implementation flow chart of function MoveCurveByKeyboard()

3 Experimental Result and Test

The testing results of the software, designed according to the above idea, is as shown in Figure 9 ~ Figure 12: Figure 9 is the demo before moving two curves; In Figure 10,when the user moves the mouse near to the curve to be translated, the color of the mouse turns red, prompt the user to move it; In Figure 11, curves make their translation through the mouse drag; In Figure 12, through the user’s clicking on the navigation key instead of using the mouse to drag, the two curves achieve translation.

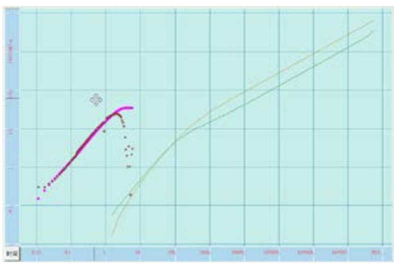


Figure 9: The Demo before Moving the Curve

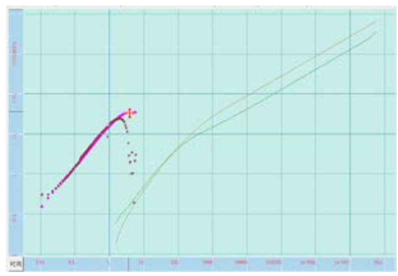


Figure 10: The Demo of Prompting the Curve Can Be Moved

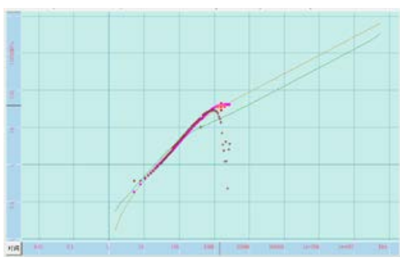


Figure 11: The Demo after Moving the Curve

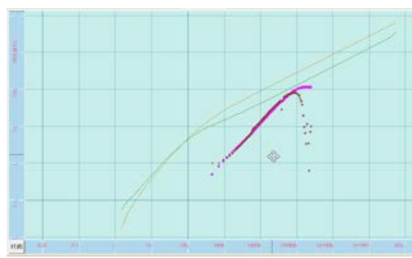


Figure 12: The Demo after Using the Keys to Move

From the above demo presentation, we can see that the software designed according to the above idea can finish the function of curve translation.

4 Conclusion

In this paper, achievements are as follows:

1) Deduce the transformation formula between the world coordinate and the screen coordinate systematically under the ordinary rectangular, the semilog and the log-log coordinate system.

2) Design of graphics translation function and calculation formula and arithmetic of world coordinate of each point after the curve translation are given.

3) Follow the design idea, in VS2010 environment, the implementation process of the software is given.

4) Take the well test interpretation software for example, a detailed testing is carried out, and the test results show that the translation function of the software runs normally and reliably.

Acknowledgements. This research is supported by national major project of China (NO 2011ZX05013). The prediction technology research of enrichment zone of relative high yield in low permeability, special low permeability reservoir. And supported by open fund of key laboratory of exploration technologies for oil and gas resources (Yangtze University), Ministry of Education (NO K2013-27).

References

[1] Alain C. Gringarten, Type-Curve Analysis: What It Can And Cannot Do, *Journal of Petroleum Technology*, **39**(1), (1987), 11-13.

[2] Gui-hua Wang, Li Xiao-ping and Zhang Jian-sheng, Software System of Interpretation for Horizontally Fractured Well's Test-Well, *Journal of Daqing Petroleum Institute*, **15**(3), (1991), 92-97.

[3] Yu Ren-yuan, Application of Modern Well Testing Interpretation Chart to Gas Well Testing Analysis, *Natural Gas Industry*, **9**(4), (1989), 38-42.

[4] Mo Ya-lin, Kou Shi-yao, Program Design Method of 2D Graphics Transformation, *Journal of Pingyuan University*, **35**(3), (1994), 63-67.

[5] Deng Yi-ming, Realization of a Multi-Functional Graphics Window, *Computer Application Research*, **12**(6), (1995), 48-49.

[6] Xu She-jiao, The Coordinate Transformation Between the two Coordinate Systems of 3D Graphics Systems, *Journal of Xidian University*, **23**(3), (1996), 429-432.

[7] Liu Yan-song, Yang Zhi-jun and Li Yuan-qi, The Introduction of Data Structure of Graphics and the Eidting of Graphics in Detail (Continued), *J. of Shenyang Institute of Aeronautical Engineering*, **14**(1), (1997), 22-28.

[8] Xie He-fang, Liu Jin-song. Geometry Conversion of the Two-Dimension Graphs, *Journal of South-Central College for Nationalities*(*Nat.Sci*.), **17**(3), (1998), 65-67.

[9] Chen Xian-ming. A Simple Method to Realize Picture Pan And Zoom in VB, *Computer Study*, **6**(6), (1999), 23-24.

[10] Yang Wei-dong and Liu Yu-shu, A Method for Converting Transformation Matrices Between Different 3D Coordinate Systems, *Journal of Computer Aided Design and Computer Graphics*, **12**(1), (2000), 53-56.

[11] Yu Wen-zhen, 2D and 3D Geometry Transformation Analysis in Computer Graphics, *Journal of Shengli Oilfield Staff University*, **15**(2), (2001), 36-37.

[12] Fan Xiao-nan, A Research of Parallel Moving for Partition Curve of Gravity Separation, *Mining And Metallurgical Engineering*, **25**(2), (2005), 25-27.

[13] Kang Hong-yan, A Solutions of Speedy Drawing, Scaling And Translation for Curves, *Computer Study*, **12**(3), (2005), 36-37.

[14] Lian Rui-mei, The Geometry Transformation and its matrix of Three-Dimension Graphics, *Journal of Weifang University*, **5**(4), (2005), 76-78.

[15] Qiao Jian-kui, Real-Time Zooming And Panning by VC Program, *Computer Programming Skills & Intenance*, **12**(10), (2005), 39-41.

[16] Li Zhan-li and Sun Xiu-ying, Realization of Coordinate Transform in VRML, *Journal of Xi'an University of Science And Technology*, **26**(2), (2006), 240-244.

[17] Ding Jian-sheng, The Research of Vector Graphics Geometric Transformation Principle Based on GDI+, *Intelligence*, **5**(34), (2010), 46-47.

[18] Ji Shu-qin,Yu Guo-feng and Zhao Hai-zhen, et al., The Discussion of Control Method of Graph in the Map Drawing Computer Software, *Computer Engineering and Applications*, **38**(11), (2003), 91-105.