한국환경분석학회지 제 2권(제 3호) 193~198, 1999
J. of the Korea Society for Environmental Analysis

3-D Windows Program for Damage Detection System of Waste Storage Land

Jae Seong Rhee*, Jong In Choe and Min Woo Jung*

Department of Chemistry, Chung-Ang University, 156-756, Korea
**Water Environmental Research Center, KIST, Seoul, 136-791, Korea*

The input data for this program are the electrical voltage differenes at 64 lattice points (8×8) of damage detection system under the multi-layer protection sheets of waste storage land. The first plot is 3 dimensional stick diagram of the intensities. The 64 intensities versus X or Y-Coordinates are fitted with multi-order polynomial and gaussian curves. The peak position of the best fitted curve is determined. These optimized two curves are combined to plot 3-dimensional surface diagram and contour diagram to visualize the calculated leak point. This computer program is written by Visual C++ 6.0. Each module is produced by one of classes of object-oriented programming language. Therefore, the maintenance and modification are fairly easy for improvement of this windows software.

1. Introduction

The bottom of massive waste storage land is covered with several layers of complex vinyl sheets to prevent the liquid solution from the garbages to penetrate the soil ground for environmental protection. But, sometimes the vinyl sheets can be damaged by sharp material, and bad liquid waste will contaminate the soil and water nearby under the land. In order to detect the exact damage location after piling up the garbages, the system which measures the electric voltage differences between above and below the complex vinyl sheets is installed before the operation of the disposal land. The electrical voltage differences at 64 lattice points (8×8) of damage detection system under the multi-layer protection sheets of proto-type sand storage pond are measured from the interface in PC. Then, this C++ program[1] read the voltage intensity data, plot them as 3-dimensional stick diagram. And the intensity versus X-coordinate is fitted as proper functions to get the accurate leak position.

Also the intensity versus Y-coordinate is fitted to determine the probable leak point.

The calculated X-Y coordinate, intensity and half-width will be used to plot 3-dimensional surfaces, and to draw 2-dimensional contour diagram which shows different colors for each level of intensity.

2. Flow Chart of Software(Plane.EXE) written by MS Visual C++ 6.0

1. Sampling Points View
1) Read input file(GreenSam.txt) of 64(8×8) points.
2) Draw stick diagram of the input intesities of sampling points.
3) Blue color means plus(+) voltage signal, red means minus(-) voltages.
4) The diagram can be rotated in 3-Dimension.

2. Fitting Intensities vs. X-coordinates to determine the X-leak position
1) Read input file(GreenSam.txt) of 64(8×8) points.

2) Read the initial values of parameters from a text file (FitCoeff.txt).
3) Draw the diagram of the intensities versus X-coordinates.
4) Calculate the best fitting parameters (coefficients) of preset functions.
   a. Gaussian function
   b. n-th order polynomial
5) Draw the fitted(calculated) curve with the experimental date.
6) Output(save) the parameters(coefficients) into the text file (FitCoeff.txt).
7) Save the X-leak positions with Intensities into another file(FitPeak.txt).

3. Fitting Intensities vs. Y-coordinates to determine the Y-leak position
1) Read input file(GreenSam.txt) of 64(8×8) points.
2) Read the initial values of parameters from a text file(FitCoefy.txt).
3) Draw the diagram of the intensities versus Y-coordinates.
4) Calculate the best fitting parameters (coefficients) of preset functions.
   a. Gaussian function
   b. n-th order polynomial
5) Draw the fitted(calculated) curve with the experimental date.
6) Output(save) the parameters(coefficients) into the text file(FitCoefy.txt).
7) Save the X and Y-leak positions with Intensities into another file(FitPeak.txt).

4. Probable Leak Peaks View of 3-D Surface
1) Read input file(FitPeak.txt) of calculated leak points.
   Format : X-coordinate, Y-coordinate, Intensity, X-width, Y-width
2) Draw 3-dimensional grid diagram of the leak intesities of X,Y positions.
3) The diagram can be rotated in 3-Dimension.

5. Contour View of the Leak Surface
1) Read input file (FitPeak.txt) of calculated leak points.
   Format : X-coordinate, Y-coordinate, Intensity, X-width, Y-width
2) Draw 2-dimensional contour diagram of the leak intensities of X,Y positions with different color for different intensity.

3. Modules in this windows software

This computer program is written by Visual C++ 6.0. Each module[2] is produced by one of classes of object-oriented programming language. Therefore, the maintenance and modification are fairly easy for improvement of this windows software.

Classes in this program:
1) CMainFrame ; Parent Window of Mutiple Document Interface
2) CChildFrame ; Child Window of Mutiple Document Interface
3) CSampling ; Sampling Points View
4) CFitView ; Fitting Intensities vs. X-coordinates to determine the X-leak position
5) CFit2View ; Fitting Intensities vs. Y-coordinates to determine the Y-leak position
6) CPlaneView ; Probable Calculated Leak Peaks 3-D Surface
7) CPlane2View ; Sampling Points 3-D Surface
8) CContourView ; Contour View of the Leak Surface

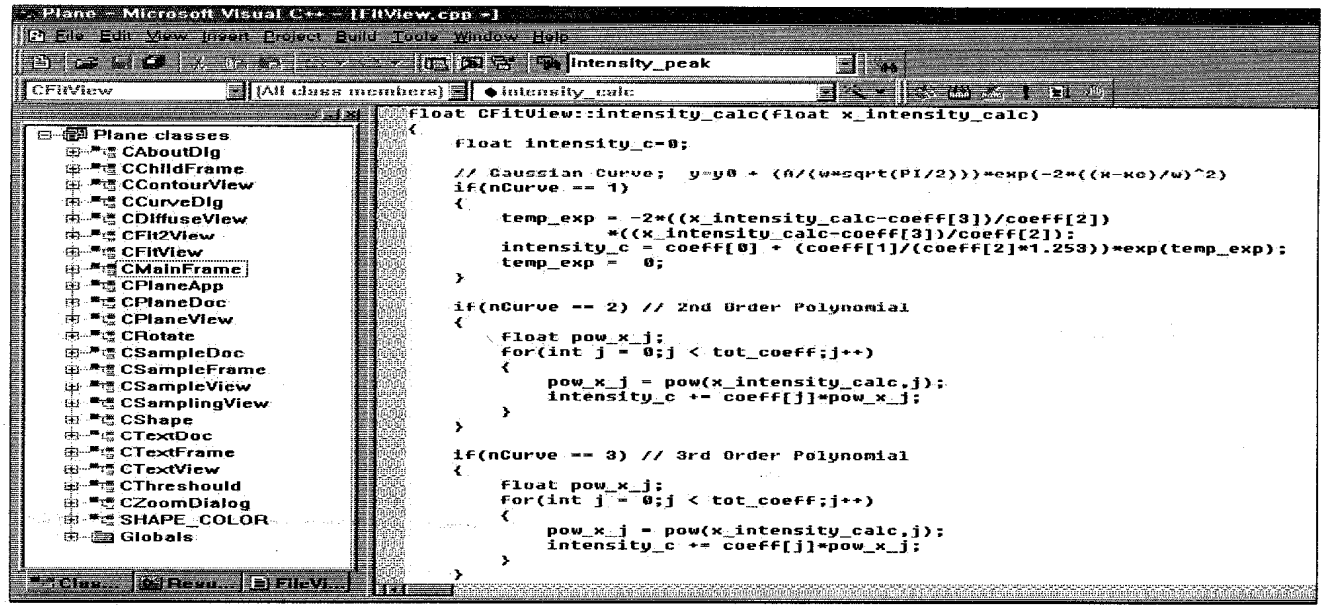


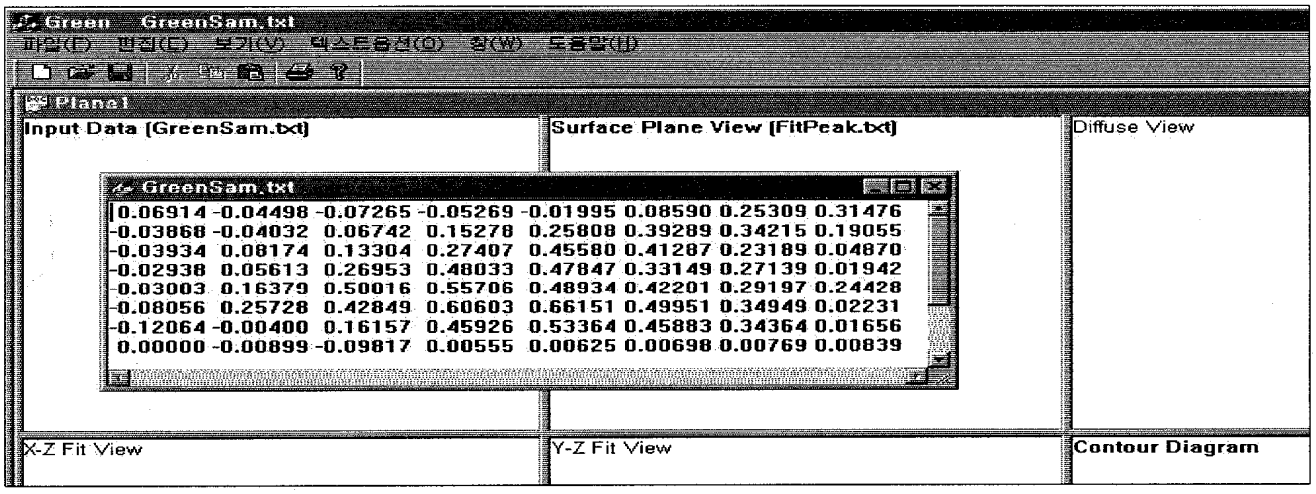
Fig. 1. Modules(Classes) in this Visual C++ software.

3. Example Source Codes for CFitView Class forr Fitting[3,4] Intensities vs. X-coordinates to determine the X-leak position

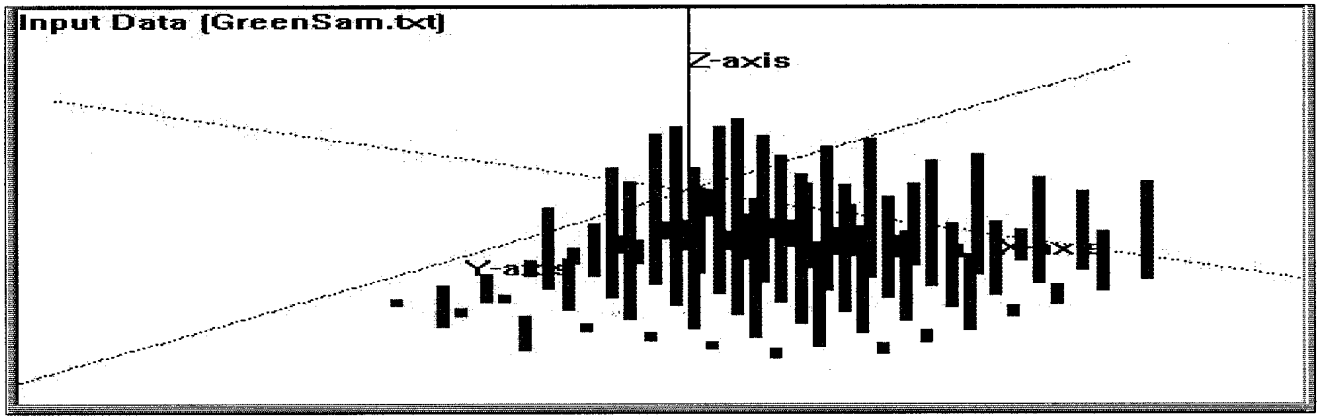
```
    load(CDC *pDC, CString filename)
          sampling[(j-1)*8 + jn].x = 0.1*j;
          sampling[(j-1)*8 + jn].y = 0.1*jn;
          sampling[j].intensity = 100*(float)sampling[j].intensity;

    Drawing3DPlane(pDC, point);
      axis(CDC* pDC)
      optimize(CDC *pDC, CPoint *point)
        residue(pDC);
          for(i = 0; i < tot_point; i++)
          {
          if(sampling[i].intensity > 0.0)
          {
          y_temp = sampling[i].y;
          intensitycalc = intensity_calc(y_temp);
          intensitysampling = sampling[i].intensity;
          deviation = intensitysampling - intensitycalc ;
          deviation2= deviation * deviation;
          new_residue += deviation2;
          }
           gradient+=grad[i]*diff(pDC, i, 1)
           gradient = sqrt(gradient);
energy_plane(pDC, point, gradient);
          x_pixel = sampling[k].x * ZOOM3D_X * ZOOMX * ratio + XCEN ;
          y_pixel = sampling[k].intensity * ZOOM3D_Y * ZOOMY + YCEN ;
          for (rx = .0; rx <= XSCAN; rx += XINTERVAL)
          {
          k = k + 1;
          j = 0;
          x_pixel = rx * ZOOM3D_X * ZOOMX * ratio + XCEN ;
          intensity_rx = intensity_calc(rx);
          if(intensity_rx > intensity_peak)
                    {
                    x_peak = rx;
                    intensity_peak = intensity_rx;
                    }
          y_pixel =  intensity_rx * ZOOM3D_Y * ZOOMY + YCEN ;
          pDC->MoveTo(x_pixel, y_pixel);
          pDC->LineTo(x_pixel+1, y_pixel);
```

4. Program Examples

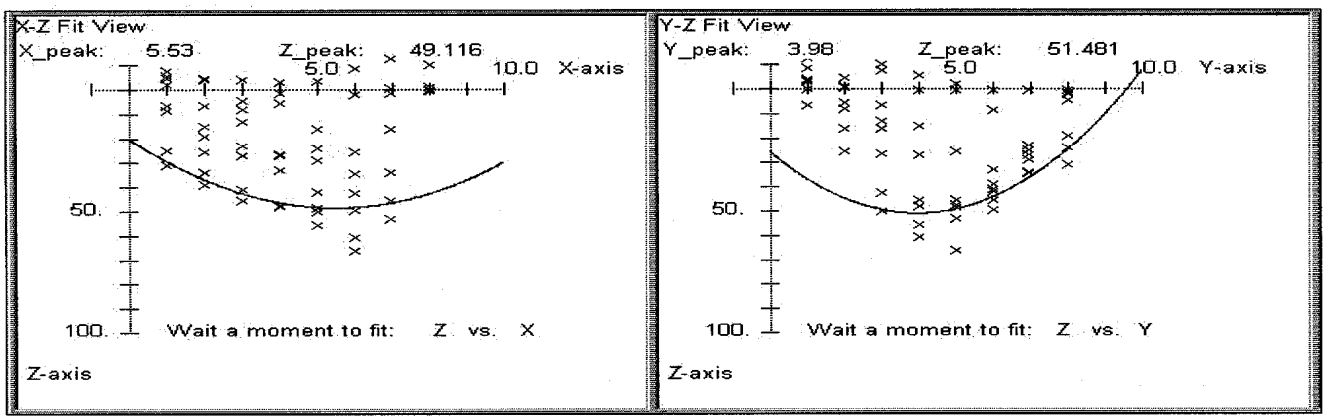
1) Input File(GreenSam.txt) and Initial Screen(Fig. 2)



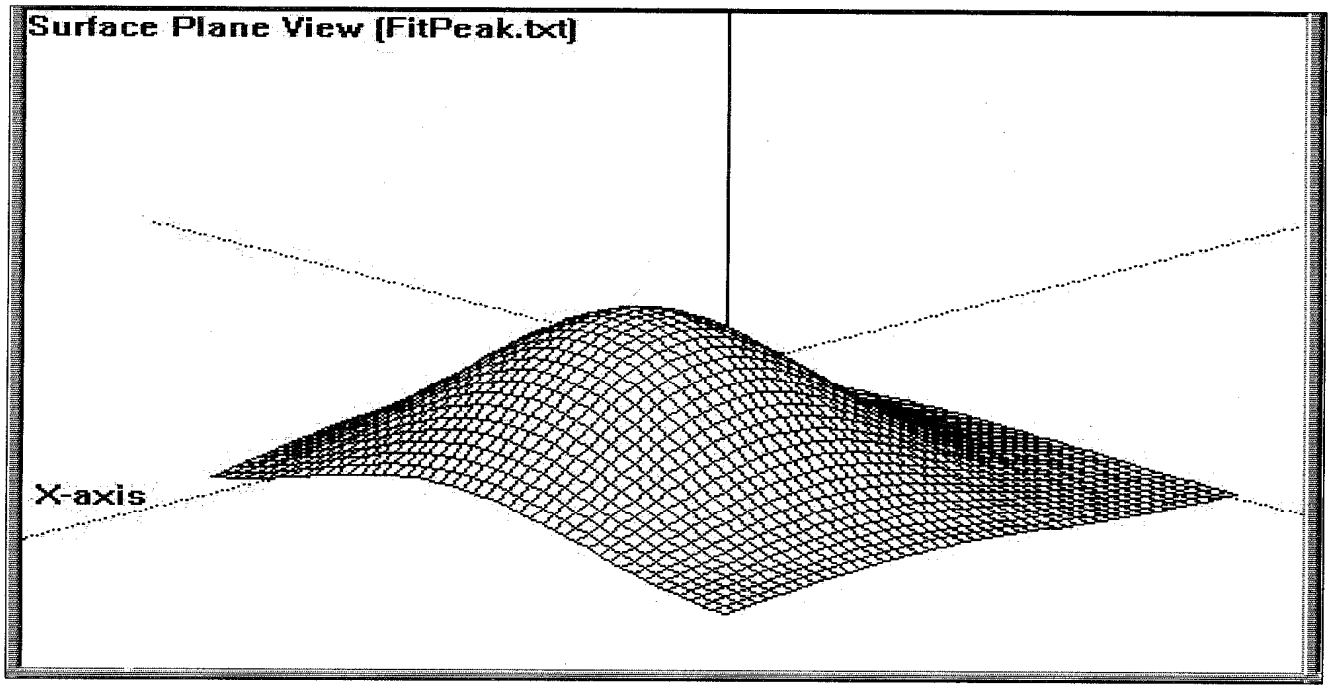
2) 3-D Stick Diagram of Input Data(Fig. 3)



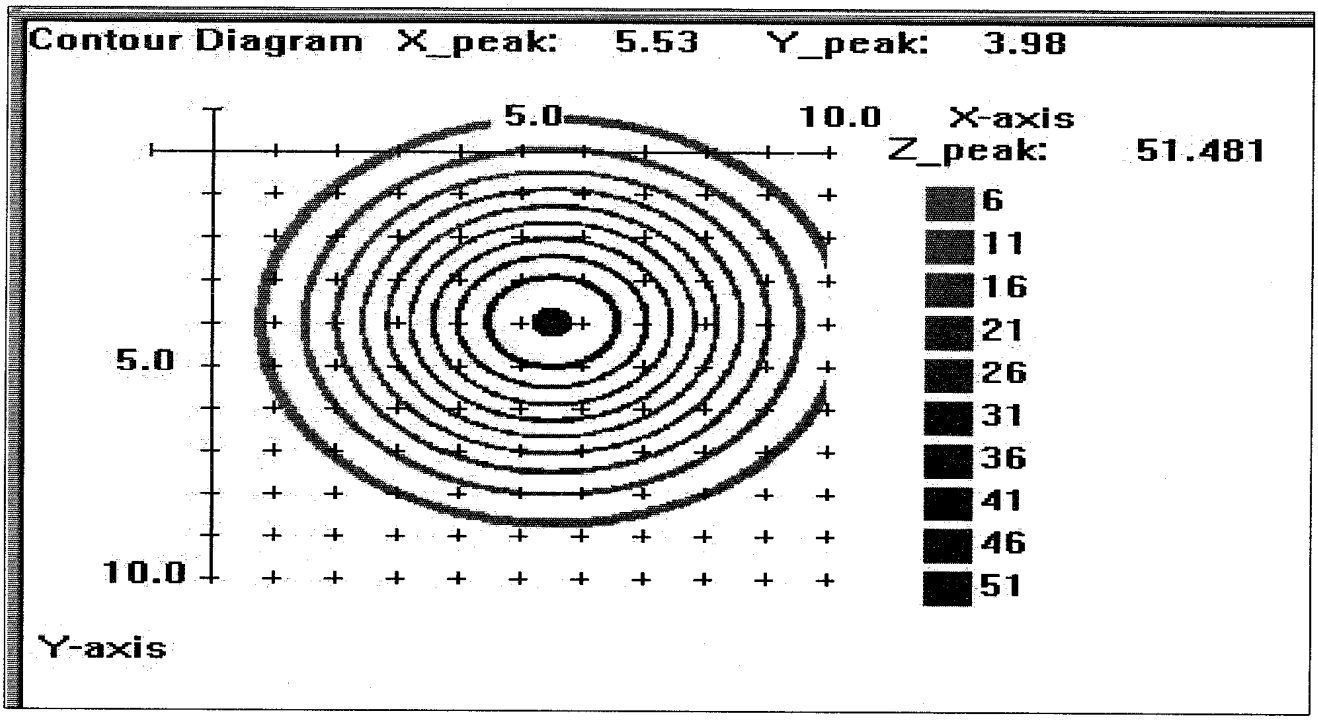
3) Fitting Views of Intensity versus X-axis and Intensity versus Y-axis(Fig. 4)



4) 3-D Surface Diagram of fitted function(Fig. 5)



5) Contour Plot of the probable intensity surface(Fig. 6)



5. Conclusion

Since this software for finding the exact location of damage under the waste storage land is written with object-oriented programming language (Visual C++), the compatibility and the extensibility are good for improvement of future upgrade.

References

1) Visual C++ 6.0 Programming Guide, **1998**, Microsoft Corporation, USA.
2) Y. S. Kim, Visual C++ 6.0 Complete Guide, **1998**, Youngjin Publishing Co., Seoul.
3) S. B. Mokhtar and C. M. Shetty, Nonlinear Programming(Theory and Algorithms), **1979**, John Wiley & Sons, New York.
4) S. L. S. Jacoby, J. S. Kowalik and J. T. Pizzo, Iterative Methods for Nonlinear Optimization Problems, **1979**, Pretice-Hall, Inc., New Jersey.