

U.S. DEPARTMENT OF THE INTERIOR  
U. S. GEOLOGICAL SURVEY

SCRNDIGW -- A WINDOWS™ Screen Digitizing program  
(an image quantitating tool)

by

Philip Powers<sup>1</sup>

Open-File Report 94-697A

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards (or with the North American Stratigraphic Code). Any use of trade, product, or firm names is for descriptive purposes only and does not imply endorsement by the U. S. Government. Although this program has been extensively tested, the U.S. Geological Survey cannot guarantee that it will give accurate results for all applications nor that It will work on all computer systems.

<sup>1</sup>U.S. Geological Survey  
Denver, CO.

# CONTENTS

	Page
Introduction .....	2
Installation.....	2
Quick users guide .....	3
Summary.....	8
References .....	9
Appendix--SCRNDIGW BASIC listing .....	10
<u>Global variables</u> .....	10
<u>Define point</u> .....	10
<u>Define pointer</u> .....	11
<u>Open file</u> .....	11
<u>Scale X</u> .....	12
<u>Scale Y</u> .....	13
<u>Picture1--main program</u> .....	14
<u>Set scale</u> .....	22

# FIGURES

Figure 1. Main screen.....	3
2. File-open menu .....	4
3. Test image.....	5
4. Define point menu .....	5
5. Define pointer menu .....	6
6. Set scale menu.....	6
7. Save file name menu.....	8

## INTRODUCTION

There have been many times that I have need to digitize curves from hardcopy plots or obtain coordinates from various features on a map. At other times I have needed to provide boundary values or pad values to a gridding program prior to contouring with a program such as SURFER by Golden Software, Inc. SCRNDIGW is a WINDOWS™ program that although simple to use is still a valuable tool for the scientist. Any image that can be put onto the WINDOWS™ clipboard can be digitized. In addition, image files of the .BMP type, .WMF type and .ICO type can also be digitized from. If you operate under WINDOWS™ and have access to a scanner, then SCRNDIGW is a tool for your software toolbag. It should be noted that if there is a fax-board in the PC, that a faxed image can be converted to a WINDOWS™ bitmap either with the fax software or a third-party program like Hijaak by Inset Systems. The converted image can then be loaded into SCRNDIGW, thus eliminating the requirement for a scanner.

This development of SCRNDIGW resulted from the improvement and conversion of the DOS-based SCRNDIG (Powers and Crosta, 92) to Microsoft's Visual BASIC™ for WINDOWS™. The minimum system requirements are a 286 processor or better, 1MB of RAM, WINDOWS™ 3.1, a mouse, a 3.5" floppy drive, and a video graphics adapter (VGA) or super VGA display.

The SCRNDIGW works with orthogonal linear coordinate systems and is equipped to correct for a slight rotation of the axes resulting from misalignment during scanning. The program does not handle map projections, however, digitizing results are satisfactory when working with small-scale maps (1:24000 or smaller). Three orthogonal points, 2 in X, and 1 in Y are required for registration and scaling. The image showing all the digitized points can be saved onto the clipboard, and then saved with the WINDOWS™ Paint program. The point size, fill color, fill type, and type of pointer are user selectable. The program has been compiled into three versions: for a standard VGA display (640 x 480), for a super VGA display (800 x 600), and for a super VGA display (1024 x 768).

The BASIC code is included for anyone who desires to improve upon the original program (see Appendix).

## INSTALLATION

To start the installation, insert the disk into the drive and copy the contents using the DOS **copy** command to a subdirectory on your hard drive. For example, from the DOS prompt at the root level, type **md scrndigw** to make a subdirectory under the root directory named scrndigw. After the program disk is inserted in your 3.5" drive, use the dos copy command, **copy a:\*. \* c:\scrndigw** to copy all the files to the subdirectory called scrndigw. The file vbrun100.dll is required to run SCRNDIRW and must reside in the same subdirectory as the executable program.

Start WINDOWS™ by typing **win**. The three most common ways to start the program are: 1) select **RUN** from the **FILE** menu and type in the drive and path and the program name, 2) double-click the name of the executable program from the **FILE MANAGER**, or 3) install an icon that can start the program by double-clicking with the mouse.

To install an icon, first select the program group that you wish SCRNDIGW to reside in and click on the **FILE** name on the command line of the Program Manager and click on **NEW.....** Next, make sure that **Program** item is selected and click on **OK**. In the **Description** input field type SCRNVGA if you have a standard 640 x 480 display, SCRN8X6 if you have an 800 x 600 display, or SCRNSVGA if your display is 1024 x 768. In the **Command Line** input field enter the name of the program (i.e. SCRNVGA.EXE). In the **Working Directory** input field enter c:\scrndigw or whatever the subdirectory has been called. To select the U.S.G.S. icon that is located in the subdirectory c:\scrndigw, click on **Change Icon** and then click on **Browse** and locate the icon file named usgs.ico and click the name to select it and then click **OK**. Then click on **OK** in the

Change Icon menu. Then click on OK on the Program Item Properties menu and the icon will appear in the selected program group.

## QUICK USERS GUIDE

The program is started by double-clicking the icon and has a main screen that appears as follows:

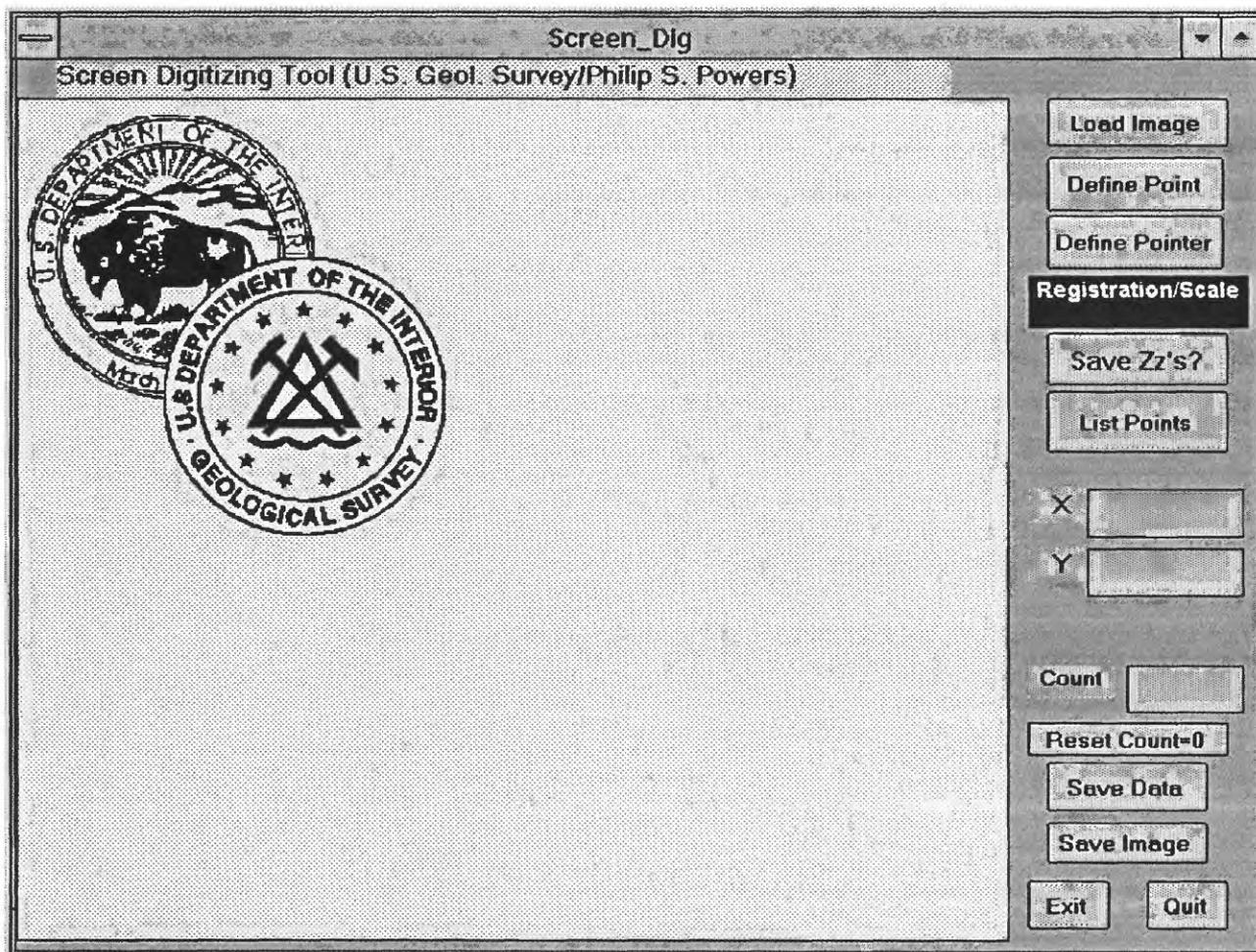


Figure 1. Main screen

The white box on the left side of the main screen, with the U.S. Geological Survey and Department of Interior emblems, is the picture box where the images will appear. Images larger than the picture box are clipped on the right and on the bottom. Pictures smaller than the picture box are placed in the upper left corner. The size of the picture box for the standard VGA display is 500 x 420 pixels. The size of the picture for the 800 x 600 display is 691 x 551 pixels. The size of the picture box for the 1024 x 768 display is 881 x 671 pixels. The buttons to the right of the

picture box are the command buttons. The sequence of command button selection is from the top to the bottom. To make a command button selection, the left mouse button is always clicked.

Starting with the first command button, click on the **Load Image** button with the left mouse button. The following menu appears:

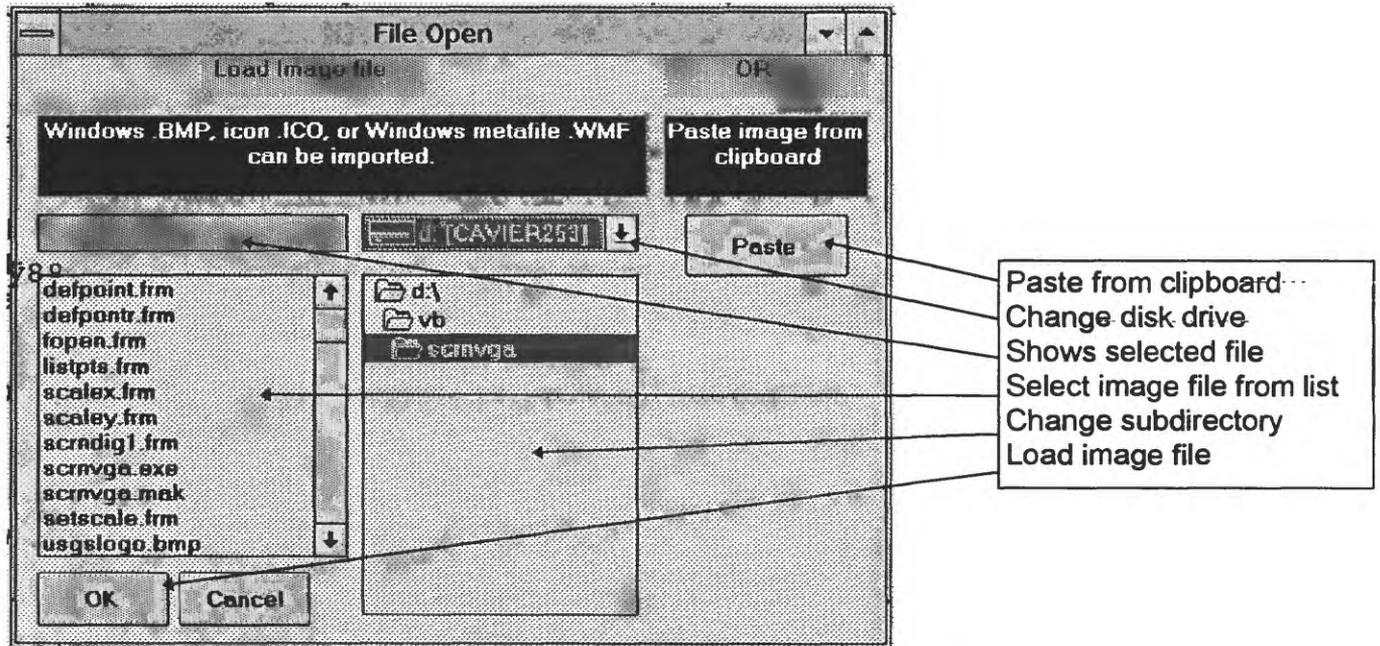


Figure 2. File-open menu.

A test image, **testimag.bmp**, (figure 3) that is ideal for testing the program is included on the program disk. Select the image by clicking **OK** and the main screen will reappear.

The appearance of the digitized points can be defined by clicking the **Define Point** command button. The following menu options (figure 4) are available to define the digitized point: There are 16 fill colors to choose from, 4 sizes from 1 to 4 pixels, and 8 fill styles. When the point type has been selected, click on the **Exit** button.

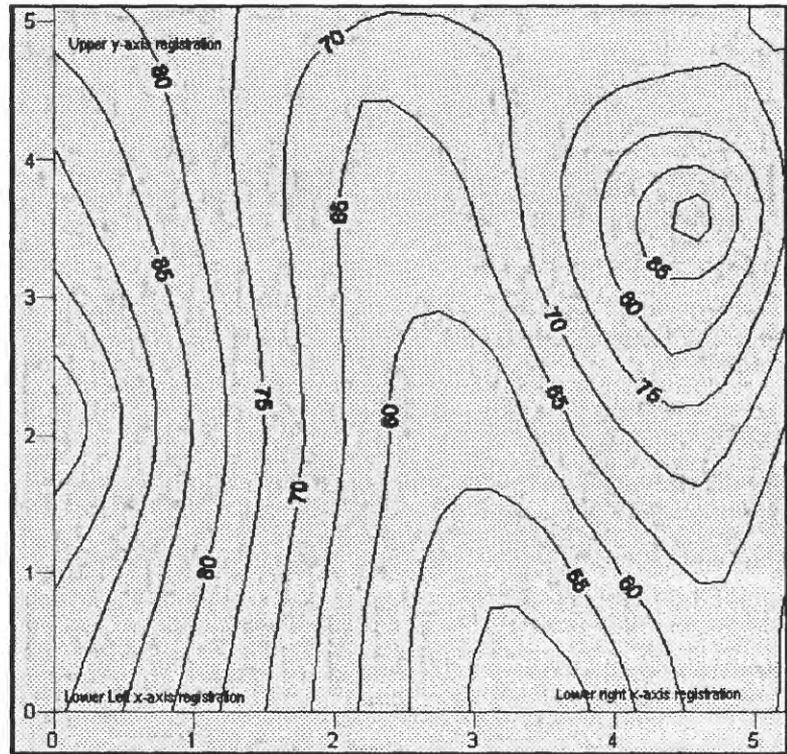


Figure 3. Test image.

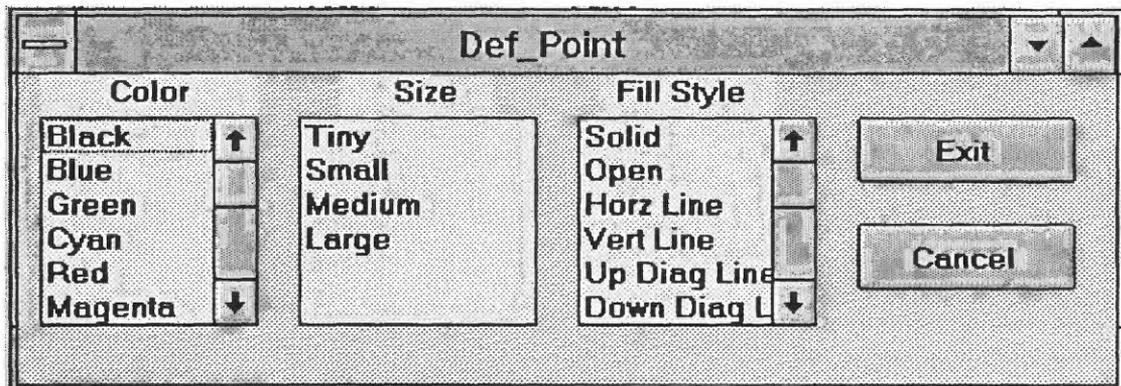


Figure 4. Define point menu.

The **Define Pointer** command button defines the pointer, and when clicked, displays the following choices:

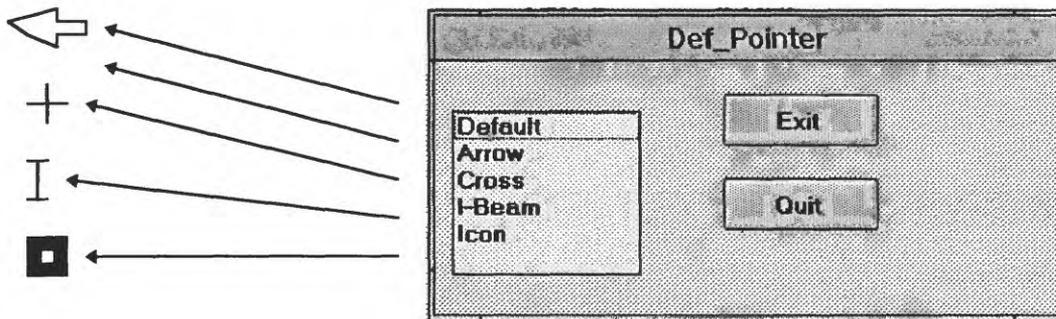


Figure 5. Define pointer menu.

The arrow (default) is most visible over dark images, while the cross is the best for precise location over an image with a light background. Click **Exit** when the pointer has been defined and the main screen re-appears.

The next command button is the registration and scale button, and is the most complex of all the command buttons. Click **Registration/Scale** and the Set Scale menu box appears as shown in figure 6:

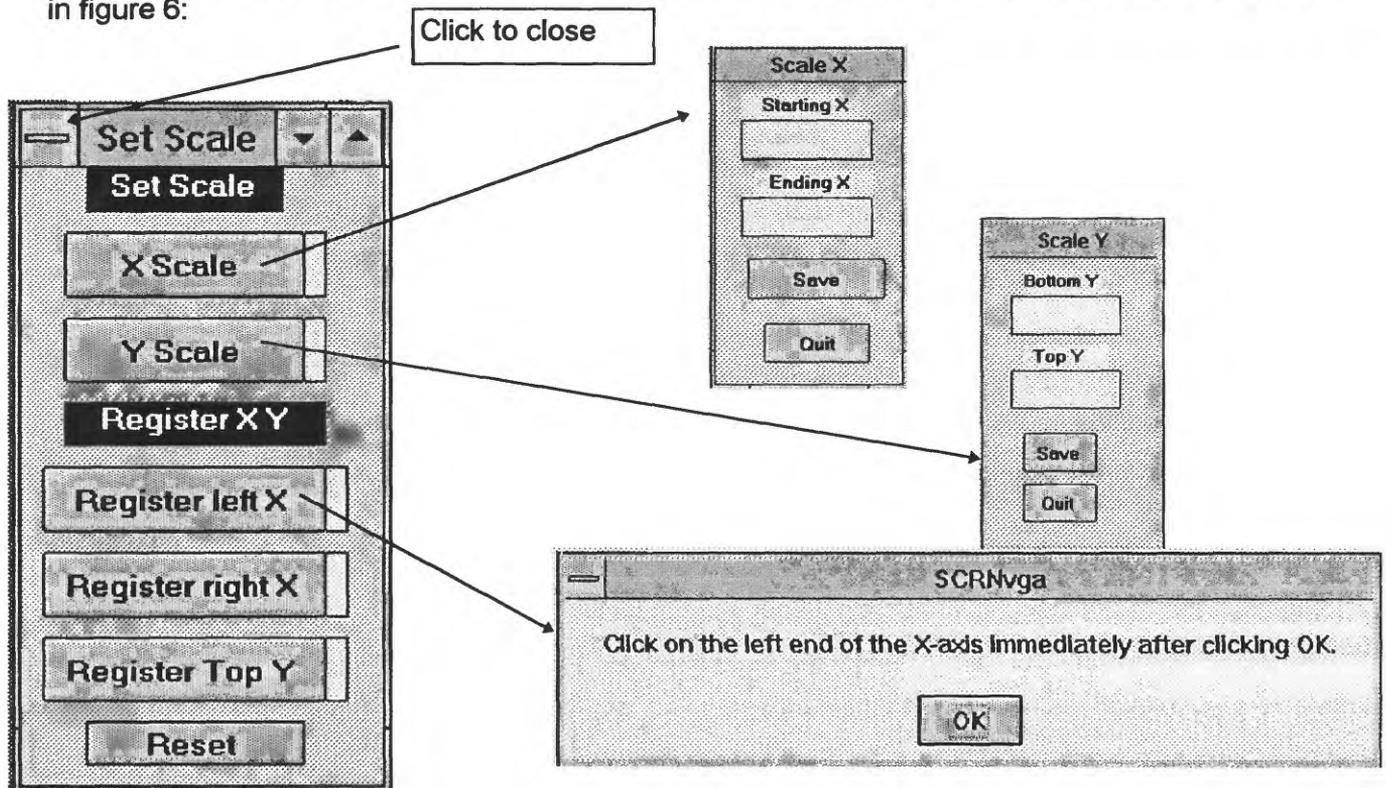
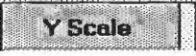
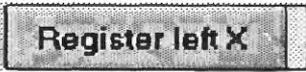


Figure 6. Set scale menu.

When the **X Scale** command button is clicked the Scale X menu box appears (as shown in figure 6). The input focus is on **Starting X**, and awaits the entry of a numerical value from the keyboard (0 in this case). The focus is shifted to **Ending X** by pressing the **tab** key and the next input value (5 in this case) is entered. Click on Save to close the Scale X menu and save the

entries. The small rectangle to the right of the button will turn red after accepting the scale values.

The  button works the same as the X Scale command button.

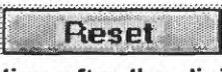
Start the registration process by clicking the  button, then click OK when prompted to. Next, position the mouse pointer on the intersection of the x-axis and y-axis (0, 0) and click the mouse. The small rectangle to the right of the Register left X button will turn red

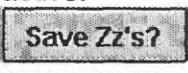
after accepting the point. Next click on the  button, click OK, and then click on the image on the x-axis and the value 5 tic mark (5, 0). To finish the registration, click the

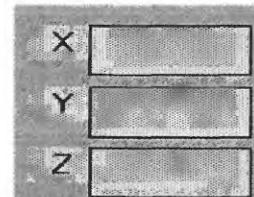
 button, click OK, and then click on the 5 tic mark on the y-axis (0, 5). The small rectangle to the right of the Register top Y button will turn blue after accepting the coordinates.

When all the rectangles are colored, click on any location in the picture box and an information box about the amount of axis rotation appears. This is an information box indicating how large of a rotation in degrees has to be made to correct the image. A negative angle is a clockwise rotation, while positive is counter-clockwise. The program mathematically corrects the image alignment. Click on OK to proceed. The Set scale menu iconized at this time. If you choose to close the Set scale menu first digitize any point in the picture box. Closing the Set scale menu before digitizing a value will result of registration.

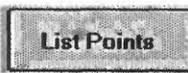
To very precisely locate the registration points the mouse can be configured with the WINDOWS™ mouse drivers to display a zoom window. To learn how to do this refer to your WINDOWS™ manual.

The  button is used to force the program to accept changes in scale or axis-registration after the digitizing process has already been started. Failure to click the reset button will leave the earlier values active.

The next command button  is for optionally saving z's. If your interest is in digitizing X and Y coordinates only, proceed to the List Points paragraph. When the Save Zz's button is clicked on, an option box that requires a Yes or No appears. If Yes is selected the command button is de-activated and the main menu is altered to show a z-input field. The focus of the program is on the z-input field, and the program will not proceed until a z-value is entered from the keyboard. Any time during the digitizing process, the focus can be shifted back to the z-input field by clicking in the field with the mouse. With the focus on the z-input field a new z-value can be entered. The value in the z-input field remains in effect until changed. At this time, the z-input field is set to take only numerical values.



X	<input type="text"/>
Y	<input type="text"/>
Z	<input type="text"/>

The next command button is . Clicking on this button produces a list of the values digitized up to this point. Because of the way the data is stored and displayed, the number of points that can be displayed is limited. After approximately 550 X, Y, and Z points have been digitized only the last ~250 points will be displayed. All the digitized values are still there but can no longer be displayed with the List Points command. The total number of points that can be digitized during one session is limited by the array size which is set to 2000. When 1950 values have been digitized, a warning message will appear advising that "Only 50 more X,Y pairs before array full.". When 2000 values have been digitized, a message appears that advises "Array full, save data or risk loss.". To save more than 2000 values, save the data to a file, and then append additional digitized values to the file with the same name.

The count-field  displays the current number of stored data pairs/triples.

The next button  resets the counter back to zero, basically erasing all the previously digitized data. This button can be used to erase test data that is used to check the registration and scale values. To clear the image of test points the image can be re-loaded,

An additional feature of the program is the capability of removing the last digitized point/points. To remove a point, position your pointer over the last digitized point, and click the right mouse button. A gray point appears over the point, and the last point is removed from the data list. This can be repeated as many times as desired. There is no intelligence built into this command, meaning that if you position the pointer over any point other than the last point, only the last point will be deleted.

The next command button is the Save Data button  and when clicked, produces the following menu box:

Click inside the input field and to the right of the drive prompt. Then type in the name of the file to save the data in. If the file already exists, you will be given three choices: 1) overwrite the data already in the file, 2) append the new data to the data already in the file, or 3) enter a new file name.

The next command button is

 and when clicked, gives the option to save the image to the clipboard or not (quit).

The last two command buttons are:

 , and will terminate the program and save the data when Exit is clicked, or just Quit the program without saving anything.

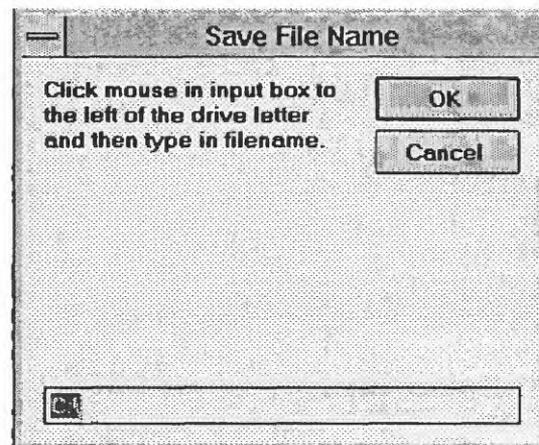


Figure 7. Save file menu.

## SUMMARY

SCRNDIGW, though lacking sophisticated features, is simple enough to be useful. Some of the applications for SCRNDIGW that I have found are: digitizing topographic profiles, digitizing contours, digitizing coordinates of cultural features on a map, digitizing images from a video capture board, and digitizing X-Y plots. SCRNDIGW can be used to gather data and teach the rudiments of digitizing without the digitizing board.

## REFERENCES

Golden Software, Inc., 1994, SURFER for Windows: USA.

Inset Systems, Inc., 1990, Hijaak graphics conversion and screen capture--User's guide: version 2.0, USA.

Microsoft Corporation, 1994, Microsoft WINDOWS™ 3.1--User's guide for the Windows graphical environment, for the MS-DOS operating system: USA.

Microsoft Corporation, 1991, Microsoft Visual Basic--Language reference: USA.

Powers, P.S., and Crosta, G., 1992, SCRNDIG--A BASIC program for digitizing from a screen image on a DOS-based computer: U.S. Geological Survey Open-File Report 92-522, 42 p.

## APPENDIX--SCRNDIGW BASIC LISTING

### Global Variables

Global File\$, Fpath\$  
Global Pfs% 'Point style  
Global psize% 'Point size  
Global Pfc% 'Point fill color  
Global Pointer% 'Type of pointer.  
Global StartX\$, EndX\$, SBottomY\$, STopY\$  
Global ct%, NextlX%, NextrX%, NextbY%, NexttY%  
Global BottomY%  
Global SbtY!, STpY!  
Global SStX!, SEdX!  
Global TopX%, TopY%, LeftY%, LeftX%, RightX%, RightY%  
Global SetDone  
Global UserX, UserY, x, y, CurrentX, CurrentY  
Global Z\$, CurZ\$, Zz, PHI, Xval1\$, Yval1\$  
Global comb\$, mouseb, paste%  
Declare Function SetWindowPos Lib "user" (ByVal h%, ByVal hb%, ByVal x%, ByVal y%, ByVal cx%, ByVal cy%, ByVal f%) As Integer  
Global Const SWP\_NOMOVE = 2  
Global Const SWP\_NOSIZE = 1  
Global Const FLAGS = SWP\_NOMOVE Or SWP\_NOSIZE  
Global Const HWND\_TOPMOST = -1  
Global Const HWND\_NOTOPMOST = -2

### Define Point

#### Sub Form\_Load ()

List1.AddItem "Black"  
List1.AddItem "Blue"  
List1.AddItem "Green"  
List1.AddItem "Cyan"  
List1.AddItem "Red"  
List1.AddItem "Magenta"  
List1.AddItem "Brown"  
List1.AddItem "White"  
List1.AddItem "Gray"  
List1.AddItem "Lt. Blue"  
List1.AddItem "Lt. Green"  
List1.AddItem "Lt. Cyan"  
List1.AddItem "Lt. Red"  
List1.AddItem "Lt. Magenta"  
List1.AddItem "Yellow"  
List1.AddItem "HI White"  
List2.AddItem "Tiny"  
List2.AddItem "Small"  
List2.AddItem "Medium"  
List2.AddItem "Large"  
list3.AddItem "Solid"  
list3.AddItem "Open"

```
list3.AddItem "Horz Line"  
list3.AddItem "Vert Line"  
list3.AddItem "Up Diag Line"  
list3.AddItem "Down Diag Line"  
list3.AddItem "Cross"  
list3.AddItem "Diag Cross"  
End Sub
```

```
Sub Command2_Click ()  
    Def_Point.Hide  
End Sub
```

```
Sub Command1_Click ()  
    If List2.listindex = 0 Then psize% = 1  
    If List2.listindex = 1 Then psize% = 2  
    If List2.listindex = 2 Then psize% = 3  
    If List2.listindex = 3 Then psize% = 4  
    Pfc% = List1.listindex  
    Pfs% = list3.listindex  
    Def_Point.Hide  
    MainScreen.Show  
End Sub
```

### **Define Pointer**

```
Sub Command1_Click ()  
    Pointer% = List1.Listindex  
    Hide  
    MainScreen.Show  
End Sub
```

```
Sub Command2_Click ()  
    Hide  
    MainScreen.Show  
End Sub
```

```
Sub Form_Load ()  
    List1.AddItem "Default"  
    List1.AddItem "Arrow"  
    List1.AddItem "Cross"  
    List1.AddItem "I-Beam"  
    List1.AddItem "Icon"  
End Sub
```

### **Open File**

```
Sub Dir1_Change ()  
    File1.Path = Dir1.Path  
End Sub
```

```
Sub Drive1_Change ()  
    Dir1.Path = Drive1.Drive
```

End Sub

**Sub File1\_Click ()**

If Len(File1.Path) = 3 Then

File\_Select.text = File1.Path + File1.FileName

Else

File\_Select.text = File1.Path + "\" + File1.FileName

End If

End Sub

**Sub Command2\_Click ()**

Hide

MainScreen.Show

End Sub

**Sub Command1\_Click ()**

If Len(File1.Path) = 3 Then

File\$ = File1.Path + File1.FileName

Else

File\$ = File1.Path + "\" + File1.FileName

End If

Fpath\$ = File1.Path

Hide

MainScreen.Show

On Error GoTo bailout:

MainScreen.Picture1.Picture = LoadPicture(File\$)

GoTo done3:

bailout:

MsgBox "There is a problem reading the image file, try a new file.", 0, "Image File Error."

FileOpen.Show

Resume done3:

done3:

End Sub

**Sub Clip\_paste\_Click ()**

Const CF\_BITMAP = 2

MainScreen.Picture1.Picture = Clipboard.GetData(CF\_BITMAP)

Hide

Paste% = 1

End Sub

### Scale X

**Sub StartX\_Change ()**

End Sub

**Sub EndingX\_Change ()**

EndingX = Val(End\_X.Text)

End Sub

**Sub EndingY\_Change ()**

End Sub

**Sub Command1\_Click ()**

StartX\$ = Start\_X.Text

EndX\$ = End\_X.Text

ScaleX.Hide

SetScale.picture1.BackColor = QBColor(12)

SetScale.Show

End Sub

**Sub Command2\_Click ()**

ScaleX.Hide

SetScale.Show

SetScale.SetFocus

End Sub

**Sub Command1\_KeyUp (KeyCode As Integer, Shift As Integer)**

StartX\$ = Start\_X.Text

EndX\$ = End\_X.Text

ScaleX.Hide

MainScreen.Show

End Sub

### Scale Y

**Sub Command1\_Click ()**

SBottomY\$ = Bottom\_Y.Text

STopY\$ = Top\_Y.Text

SBtY = Val(SBottomY\$)

STpY = Val(STopY\$)

SetScale.picture2.BackColor = QBColor(12)

SetScale.Show

ScaleY.Hide

End Sub

**Sub Command2\_Click ()**

SetScale.Show

ScaleY.Hide

End Sub

**Sub Command1\_KeyPress (KeyAscii As Integer)**

SBottomY\$ = Bottom\_Y.Text

STopY\$ = Top\_Y.Text

SBtY = Val(SBottomY\$)

STpY = Val(STopY\$)

Hide

MainScreen.Show

End Sub

Dim DrawNow As Integer

Dim ArrayX(2000), ArrayY(2000), ArrayZ(2000)

**Picture1--main program**

**Sub Picture1\_Click ()**

Scale (0, 10000)-(10000, 0)

'In case Fill color < 0 set to 0

If Pfc% < 0 Then Pfc% = 0

'In case Fill Style < 0 set to 0

If Pfs% < 0 Then Pfs% = 0

'In case psize%<0 then set to 2

If psize% < 0 Then psize% = 2

'Disable Saving Zz's Command button if the counter is already 1 or more.

If ct% > 0 Then Command4.enabled = 0

If mouseb = 2 Then 'right mouse button pressed.

'decrement the counter by one and mark the last point as erased.

ct% = ct% - 1

Picture1.FillColor = QBColor(8)

Picture1.FillStyle = 0

Picture1.Circle (CurrentX, CurrentY), 2, QBColor(8)

count.Text = Format\$(ct%)

GoTo around:

End If

NL\$ = Chr\$(13) + Chr\$(10)

If Len(CurZ\$) = 0 And Z\$ = "Y" Then

MsgBox "Enter a value in the Z-field immediately after clicking OK"

End If

Picture1.FillColor = QBColor(Pfc%)

Picture1.FillStyle = Pfs%

Picture1.Circle (CurrentX, CurrentY), psize%, QBColor(0)

' Left X

If NextIX% = 1 Then

LeftX% = CurrentX

LeftY% = CurrentY

SetScale.picture3.BackColor = QBColor(12)

NextIX% = 0

GoTo around:

End If

'Right X

```
If NextrX% = 1 Then
  RightX% = CurrentX
  rightY% = CurrentY
  SetScale.picture4.BackColor = QBColor(12)
  NextrX% = 0
  GoTo around:
End If
```

'Top of Y

```
If NexttY% = 1 Then
  TopY% = CurrentY
  TopX% = CurrentX
  SetScale.picture5.BackColor = QBColor(9)
  NexttY% = 0
  GoTo around:
End If
```

If SetDone = 1 Then GoTo skip:

```
If SetScale.picture3.BackColor = QBColor(12) And SetScale.picture4.BackColor = QBColor(12)
And SetScale.picture5.BackColor = QBColor(9) Then
```

'Angle of rotation (+ is CCW, - is CW)

```
PHI = Atn((LeftY% - rightY%) / (RightX% - LeftX%))
```

```
pi = 3.141593
```

```
tmp = (180 / pi) * PHI
```

```
MsgBox "Image is rotated " + Str$(tmp) + " degrees. Program automatically corrects."
```

'Angle between X and Y axes.

```
phi2 = Atn(Abs(TopX% - LeftX%) / Abs(TopY% - LeftY%))
```

```
lenX = Abs(RightX% - LeftX%) / Cos(PHI)
```

```
lenY = Abs(TopY% - LeftY%) / Cos(PHI)
```

'inches per screen unit

```
If SetScale.Picture1.BackColor = QBColor(12) And SetScale.picture2.BackColor = QBColor(12)
Then
```

Rem Scale factors for X and Y have been entered.

```
SStX = Val(StartX$)
```

```
SEdX = Val(EndX$)
```

```
xrange = Abs(SEdX - SStX)
```

```
yrange = Abs(STpY - SBTY)
```

```
UserX = lenX / (xrange)
```

```
UserY = lenY / (yrange)
```

```
SetDone = 1
```

End If

End If

skip:

If SetDone = 1 Then

A = CurrentX - LeftX%

B = LeftY% - CurrentY

amd2:

ct% = ct% + 1

corrX = A \* Cos(PHI) + B \* Sin(PHI)

'An attempt to correct for the case where the ending X is smaller than the starting X

If SEdX < SStX Then corrX = -corrX

corrY = B \* Cos(PHI) - A \* Sin(PHI)

Xval = (SStX + (corrX / UserX))

Xloc.Text = Format\$(Xval, "#####.#####")

Yval = ((corrY / UserY) + SBtY)

Yloc.Text = Format\$(Yval, "#####.#####")

count.Text = Format\$(ct%)

ArrayX(ct%) = SStX + (corrX / UserX)

ArrayY(ct%) = SBtY + (corrY / UserY)

ArrayZ(ct%) = Zz

If Len(comb\$) > 20000 Then

MsgBox "Caution: List Points XYZ values will display only most recently digitized points. "

comb\$ = Right\$(comb\$, 10000)

End If

If Z\$ = "Y" Then

XYZ\$ = Str\$(ct%) + " " + Str\$(ArrayX(ct%)) + " " + Str\$(ArrayY(ct%)) + " " +

Str\$(ArrayZ(ct%)) + NL\$

comb\$ = comb\$ + XYZ\$

GoTo listdone:

End If

XY\$ = Str\$(ct%) + " " + Str\$(ArrayX(ct%)) + " " + Str\$(ArrayY(ct%)) + NL\$

comb\$ = comb\$ + XY\$

listdone:

End If

Screen.MousePointer = Pointer%

around:

End Sub

**Sub Picture1\_MouseDown (Button As Integer, Shift As Integer, X As Single, Y As Single)**

DrawNow = -1

Scale (0, 10000)-(10000, 0)

CurrentX = X

CurrentY = Y

mouseb = Button

End Sub

```
Sub Picture1_MouseUp (Button As Integer, Shift As Integer, X As Single, Y As Single)  
    DrawNow = 0  
End Sub
```

```
Sub Picture1_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)  
    Scale (0, 10000)-(10000, 0)  
    If DrawNow Then PSet (X, Y)  
End Sub
```

```
Sub Save_Image_Click ()  
    FileOpen.Show  
End Sub
```

```
Sub Load_Image_Click ()  
    SetScale.Scale_X.enabled = -1  
    SetScale.Scale_Y.enabled = -1  
    SetScale.Reg_IX.enabled = -1  
    SetScale.Reg_rX.enabled = -1'Register right end of x axis.  
    SetScale.Reg_tY.enabled = -1'Register top end of y axis.  
    paste% = 0  
    FileOpen.Show  
End Sub
```

```
Sub Command1_Click ()  
Picture1.Picture = Picture1.Image  
Choice1 = MsgBox("Click OK to paste to clipboard.", 1, "Save Choice")  
Select Case Choice1  
    Case 1  
        Clipboard.SetData (Picture1.Picture)  
        MsgBox "Image copied to the Clipboard."  
    Case 2  
        GoTo done2:  
    Case 6  
        temp1$ = InputBox$("Enter the output filename for the image.", "Save File Name", File$)  
        If temp1$ = "" Then  
            GoTo done2: 'CANCEL button pressed.  
        End If  
        SavePicture Picture1.Image, temp1$  
  
End Select  
  
done2:  
End Sub
```

```
Sub Command3_Click ()  
    comb$ = ""  
    count.Text = "0"  
    Xloc.Text = ""  
    Yloc.Text = ""  
    ct% = 0  
    Command4.enabled = 1
```

End Sub

**Sub Command5\_Click ()**

End

End Sub

**Sub Command2\_Click ()**

NexttY% = 1

MsgBox "Click on the Top end" + Chr\$(13) + "of the Y-axis " + Chr\$(13) + "immediately after clicking OK"

End Sub

**Sub Form\_Click ()**

Screen.MousePointer = Pointer%

End Sub

**Sub Command7\_Click ()**

Rem Hide

ScaleX.Show

End Sub

**Sub Count\_Change ()**

If Val(count.Text) > 1950 Then

MsgBox "Only 50 more X,Y pairs before array full.", 48, "WARNING"

End If

If Val(count.Text) = 2000 Then

MsgBox "Array full, save data or risk loss.", 48, "DANGER"

End If

End Sub

**Sub Command4\_Click ()**

Response% = MsgBox("Do you wish to save Zz's?", 4, "Save Z")

If Response% = 6 Then

Zzs.visible = -1

Label8.visible = -1

Z\$ = "Y"

Command4.enabled = 0

Zzs.SetFocus

Else

Z\$ = ""

Zzs.visible = 0

Label8.visible = 0

End If

End Sub

**Sub Define\_point\_Click ()**

Def\_Point.Show

End Sub

```
Sub Define_pointer_Click ()
    Def_Pointer.Show
End Sub
```

```
Sub Finished_Click ()
    Call Save_data_Click
    Call Command1_Click
End
End Sub
```

```
Sub Quit_Click ()
    End
End Sub
```

```
Sub Save_data_Click ()
start:
```

'The following If skips the 1st round of file/path input.

If Len(tempname\$) > 0 Then GoTo around1:

If Len(Fpath\$) = 3 Then

    Opath\$ = Fpath\$ + tempname\$

Else

    Opath\$ = "C:\\"

End If

around1:

tempname1\$ = ""

On Error GoTo ermsg1:

If Len(tempname\$) > 0 Then

    tempname1\$ = tempname\$

    tempname\$ = InputBox\$("Click mouse in input box to the left of the drive letter and then type in the filename.", "Save File Name", tempname\$)

If tempname\$ = "" Then

    tempname\$ = tempname1\$

    GoTo done1:

End If

Else

    tempname\$ = InputBox\$("Click mouse in input box to the left of the drive letter and then type in filename.", "Save File Name", Opath\$)

End If

If tempname\$ = Opath\$ Then

    MsgBox "You must enter a filename."

    GoTo around1:

End If

If tempname\$ = "" Then

    GoTo done1: 'CANCEL button pressed.

End If

Open tempname\$ For Input As #2

Response% = MsgBox("File exists, Yes to overwrite, No to append, Cancel to enter another name.", 3, "WARNING")

```
Select Case Response%
  Case 6 'Overwrite
    Close #2
    Open tempname$ For Output As #2
    Resume dump
  Case 2 'Cancel
    Close #2
    GoTo start:
  Case 7 'Append
    Close #2
    Open tempname$ For Append As #2
    Resume dump:
End Select
```

```
dump:
  If Z$ = "Y" Then
    For i% = 1 To ct%
      Print #2, ArrayX(i%), ArrayY(i%), ArrayZ(i%)
    Next i%

    GoTo dispdone:
  End If
```

```
  For i% = 1 To ct%
    Print #2, ArrayX(i%), ArrayY(i%)
  Next i%
```

```
dispdone:
  MsgBox "Data Written to File."
  GoTo done1:
```

```
errmsg1:
  On Error GoTo 0
  Close #2
  Open tempname$ For Output As #2
  Resume dump:
```

```
done1:
  Close #2
```

End Sub

```
Sub X_Scale_Click ()
  ScaleX.Show
End Sub
```

```
Sub Scale_X_Click ()
```

End Sub

```

Sub Reg_IX_Click ()
  NextIX% = 1
  MsgBox "Click on the left end of the X-axis immediately after clicking OK"
  Scale (0, 10000)-(10000, 0)
End Sub

```

```

Sub Register_Y_Click ()
  NextbY% = 1
  MsgBox "Click on the Bottom end" + Chr$(13) + "of the Y-axis " + Chr$(13) + "immediately after clicking OK"
End Sub

```

```

Sub Y_Scale_Click ()
  ScaleY.Show
End Sub

```

```

Sub X_Scale_GotFocus ()
' Is the X scale set?
'If so, indicate with a red mark by control button.

  If (StartX$) <> "" Then
    SetScale.Picture1.BackColor = QBColor(12)
  End If

End Sub

```

```

Sub Y_Scale_GotFocus ()
' Is the Y scale set?
'If so, indicate with a red mark by control button.

  If STopY$ <> "" Then
    SetScale.picture2.BackColor = QBColor(12)
  End If

End Sub

```

```

Sub Zzs_Change ()
  CurZ$ = Zzs.Text
  Zz = Val(Zzs.Text)
End Sub

```

```

Sub Label6_Click ()
  success% = SetWindowPos(SetScale.hWnd, HWND_TOPMOST, 0, 0, 0, 0, FLAGS)
  Rem success%<>0 When Successful
  SetScale.Show
End Sub

```

```

Sub List_Pts_Click ()
  ListPts.Text1.Text = comb$
  ListPts.Show
End Sub

```

## Set Scale

```
Sub Redo1_Click ()  
    SetDone = 0  
End Sub
```

```
Sub Scale_X_Click ()  
    ScaleX.Show  
    ScaleX.Start_X.SetFocus  
End Sub
```

```
Sub Scale_Y_Click ()  
    ScaleY.Show  
    ScaleY.Bottom_Y.SetFocus  
End Sub
```

```
Sub Reg_IX_Click ()  
    NextIX% = 1  
    MsgBox "Click on the left end of the X-axis immediately after clicking OK."  
    SetScale.Show  
End Sub
```

```
Sub Reg_rX_Click ()  
    NextX% = 1  
    MsgBox "Click on the right end of the X-axis immediately after clicking OK."  
    SetScale.Show  
End Sub
```

```
Sub Reg_tY_Click ()  
    NextY% = 1  
    MsgBox "Click at the Top of the Y-axis immediately after clicking OK."  
    SetScale.Show  
End Sub
```

```
Sub Form_Unload (Cancel As Integer)  
    success% = SetWindowPos(SetScale.hWnd, HWND_NOTOPMOST, 0, 0, 0, 0, FLAGS)  
    Rem success% <> 0 When Successful  
End Sub
```

```
Sub Redo1_Click ()  
    SetDone = 0  
End Sub
```