SpaceResection

软件说明书

2018011班第11组

201801131常志鹏

201801132黄义超

201801133韩 洋

201801134廉年刚

1. **引言**

**1.1编写目的**

在航空摄影测量中，如果确定了每幅影像的9个内外方位元素，就能确定被摄物体与航摄影像之间的解析关系，为制图奠定基础。利用影像范围内一定数量的控制点的空间坐标以影像坐标，根据共线条件方程，采用间接平差，求解该影像方位元素的方法，称为单幅影像的空间后方交会法。本软件求解单幅影像的外方位元素。

**1.2项目背景**

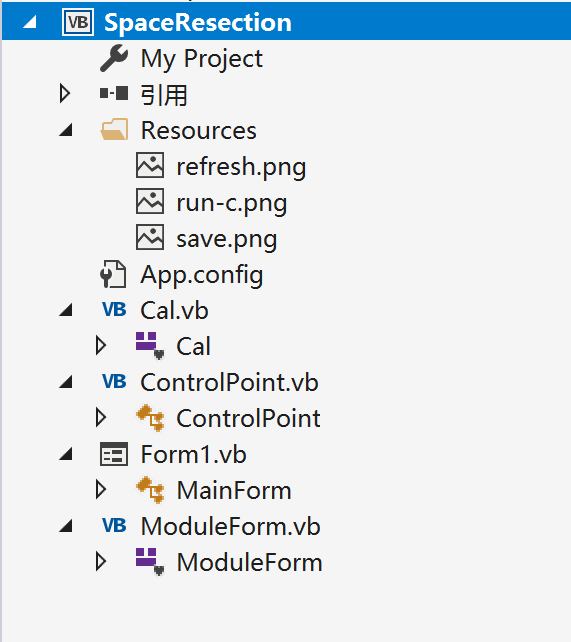
软件系统名称：SpaceResection

软件开发平台：Visual Studio 2015

该软件运行于Windows系统平台

1. **总体设计**

**2.1软件结构**

****

**2.2软件原理**

以单幅影像为基础，从该影像所覆盖地面范围内的若干控制点的已知地面坐标和相应点的像坐标量测值出发，根据共线条件方程，运用最小二乘间接平差，求解该影像在航空摄影时刻的外方位元素。

由于共线方程是非线性函数，为了运用最小二乘法，必须先将其线性化。

共线条件方程的线性化



某一点的共线方程为：

式中，x,y 为这一点的像平面坐标，x0,y0,f 为影像的内方位元素， Xs,Ys,Zs 为摄站点的物方空间坐标， X,Y,Z 为这一点的物方空间坐标。 ai,bi,ci(i=1,2,3) 为影像旋转矩阵的九个元素，即：



因为未知数是外方位元素，所以将共线方程视为外方位元素的函数。设外方位元素的近似值为



将共线方程在外方位元素近似值处一阶泰勒展开，得：

式中 x0,y0 是把外方位元素近似值代入共线方程中得到的 x,y 。

列出误差方程

将控制点对应的像点的像平面坐标视为观测值，外方位元素视为参数，由共线方程的线性形式可列出误差方程：

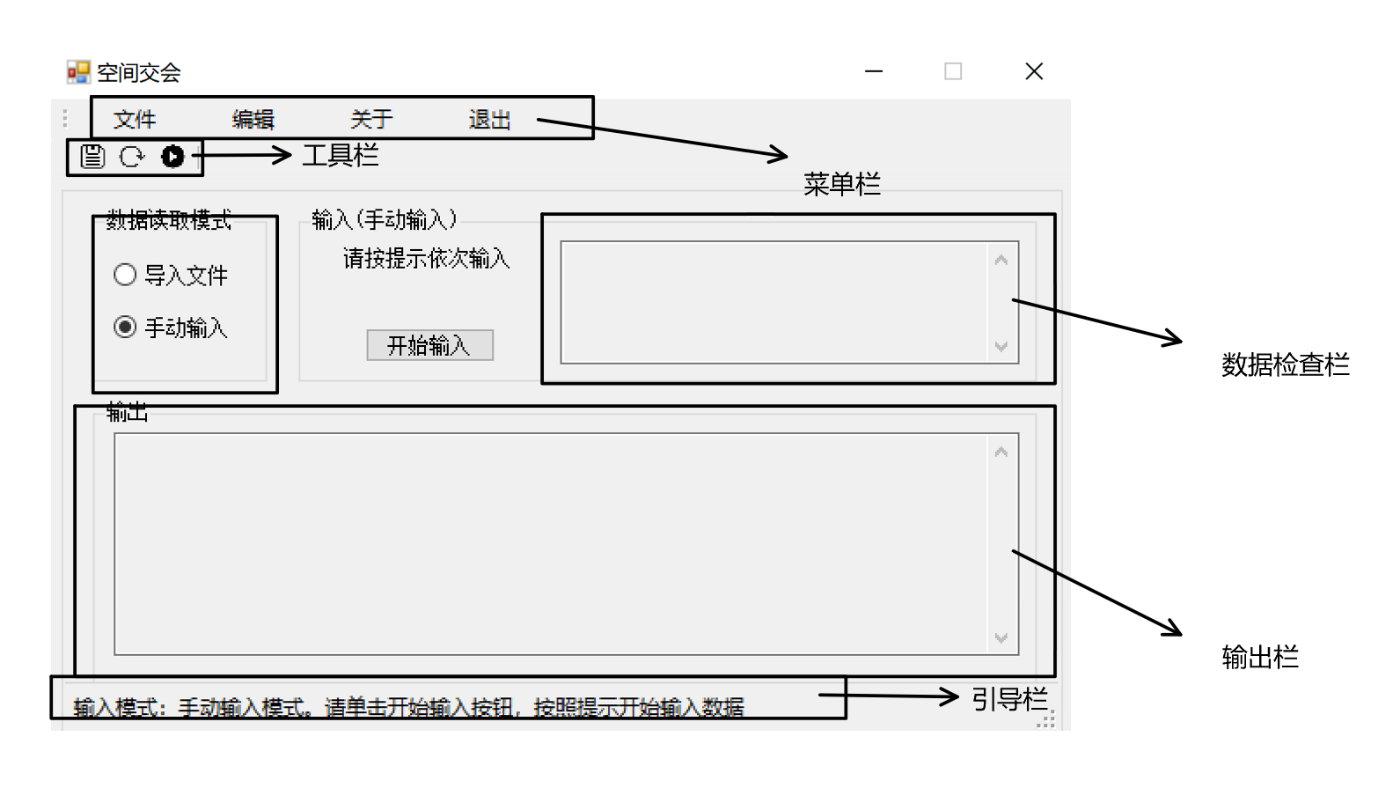
式中 x,y 为控制点对应的像点的像平面坐标，Vx,Vy 为像平面坐标的改正数,

为参数的改正数。

以上两个方程为一个控制点列出。如果有 n 个控制点，则可以列出 2n 个方程。当 n>=3 时就可求解。

1. **程序描述**

**3.1界面描述**

****

**3.2程序描述**

本软件通过手动输入和文件导入两种模式导入数据，计算出相片的外方位元素，并给出误差方程。

1. **使用说明**

**4.1手动输入模式**

选择数据读取模式为手动输入模式，在输入窗口选择开始输入按钮，在出现的文本框中输入摄影机主距f：，单击确认按钮。输入在文本框中再次输入控制点1的信息，格式为“格式：控制点号,像点x,像点y,物点X,物点Y,物点Z”，单击确认按钮。依次输入四个控制点的信息，输入数据将在检查窗口实时出现，如有错误，单击工具栏中间按钮重新开始输入。输入数据完毕后，单击开始计算按钮或工具栏第三个按钮，此时输出窗口会显示输入数据，计算出的外方位元素及误差方程。如需保存，点击工具栏第一个按钮将输出内容保存。

**4.2文件导入模式**

选择数据读取模式为文件导入模式，在输入窗口选择开始导入按钮，导入数据将在检查窗口实时出现，如有错误，请检查文件，并单击工具栏中间按钮重新开始导入。导入数据完毕后，单击开始计算按钮或工具栏第三个按钮，此时输出窗口会显示输入数据，计算出的外方位元素及误差方程。如需保存，点击工具栏第一个按钮将输出内容保存。

1. **程序文件**

Cal.vb

Imports System.Math

Module Cal  *'计算外方位元素*

    Public RepeatNumber As Integer = 0, m As Integer = 40000, GCPNUMBER As Integer = 4

    Public x0 As Double = 0, y0 As Double = 0

    Public **Unknownum**(5) As Double *'外方位元素*

    Public **UnKnown**(5, 5) As Double *'误差方程*

    Public **UnKnow**() As Double = {1, 1, 1, 1, 1, 1}   *'外方位元素改正数*

    Public NN As Integer = 6   *'阶数*

    Public f As Double *'摄影机主距*

    Sub **Calculat**()

        Dim **Xxy**(GCPNUMBER - 1, 1) As Double  *'存放控制点像点坐标*

        Dim **WXYZ**(GCPNUMBER - 1, 2) As Double *'存放控制点物点坐标*

        Dim i%, j%, k%

        For i = 0 To GCPNUMBER - 1

            For j = 0 To 1

**Xxy**(i, j) = **Val**(DataStr$(6 \* i + j + 2))

            Next

        Next

        For i = 0 To GCPNUMBER - 1

            For j = 0 To 2

**WXYZ**(i, j) = **Val**(DataStr$(6 \* i + j + 4))

            Next

        Next

        Dim **sunXYZ**(3) As Double      *'物点XYZ平均值*

        For i = 0 To 2

            For j = 0 To GCPNUMBER - 1

**sunXYZ**(i) += **WXYZ**(j, i)

            Next

            For j = 0 To 1

**Unknownum**(i) = **sunXYZ**(i) / GCPNUMBER *'XY初始值*

            Next

        Next

**Unknownum**(2) = m \* f + **sunXYZ**(2) / GCPNUMBER *'Z初始值*

**Unknownum**(3) = 0  *'外方位初始值*

**Unknownum**(4) = 0

**Unknownum**(5) = 0

        Do

            Dim **R**(2, 2) As Double  *'旋转矩阵*

**R**(0, 0) = **Cos**(**Unknownum**(3)) \* **Cos**(**Unknownum**(5)) - **Sin**(**Unknownum**(3)) \* **Sin**(**Unknownum**(4)) \* **Sin**(**Unknownum**(5))

**R**(0, 1) = -**Cos**(**Unknownum**(3)) \* **Sin**(**Unknownum**(5)) - **Sin**(**Unknownum**(3)) \* **Sin**(**Unknownum**(4)) \* **Cos**(**Unknownum**(5))

**R**(0, 2) = -**Sin**(**Unknownum**(3)) \* **Cos**(**Unknownum**(4))

**R**(1, 0) = **Cos**(**Unknownum**(4)) \* **Sin**(**Unknownum**(5))

**R**(1, 1) = **Cos**(**Unknownum**(4)) \* **Cos**(**Unknownum**(5))

**R**(1, 2) = -**Sin**(**Unknownum**(4))

**R**(2, 0) = **Sin**(**Unknownum**(3)) \* **Cos**(**Unknownum**(5)) + **Cos**(**Unknownum**(3)) \* **Sin**(**Unknownum**(4)) \* **Sin**(**Unknownum**(5))

**R**(2, 1) = -**Sin**(**Unknownum**(3)) \* **Sin**(**Unknownum**(5)) + **Cos**(**Unknownum**(3)) \* **Sin**(**Unknownum**(4)) \* **Cos**(**Unknownum**(5))

**R**(2, 2) = **Cos**(**Unknownum**(3)) \* **Cos**(**Unknownum**(4))

            Dim **Xgang**(GCPNUMBER - 1) As Double

            For i = 0 To GCPNUMBER - 1

**Xgang**(i) = **R**(0, 0) \* (**WXYZ**(i, 0) - **Unknownum**(0)) + **R**(1, 0) \* (**WXYZ**(i, 1) - **Unknownum**(1)) + **R**(2, 0) \* (**WXYZ**(i, 2) - **Unknownum**(2))

            Next

            Dim **Ygang**(GCPNUMBER - 1) As Double

            For i = 0 To GCPNUMBER - 1

**Ygang**(i) = **R**(0, 1) \* (**WXYZ**(i, 0) - **Unknownum**(0)) + **R**(1, 1) \* (**WXYZ**(i, 1) - **Unknownum**(1)) + **R**(2, 1) \* (**WXYZ**(i, 2) - **Unknownum**(2))

            Next

            Dim **Zgang**(GCPNUMBER - 1) As Double

            For i = 0 To GCPNUMBER - 1

**Zgang**(i) = **R**(0, 2) \* (**WXYZ**(i, 0) - **Unknownum**(0)) + **R**(1, 2) \* (**WXYZ**(i, 1) - **Unknownum**(1)) + **R**(2, 2) \* (**WXYZ**(i, 2) - **Unknownum**(2))

            Next

            Dim **approxxy**(GCPNUMBER - 1, 1) As Double

            For i = 0 To GCPNUMBER - 1

**approxxy**(i, 0) = -f \* **Xgang**(i) / **Zgang**(i)

**approxxy**(i, 1) = -f \* **Ygang**(i) / **Zgang**(i)

            Next

            Dim **A**(GCPNUMBER \* 2 - 1, 5) As Double    *'A*

            For i = 0 To GCPNUMBER - 1

**A**(i \* 2, 0) = (**R**(0, 0) \* f + **R**(0, 2) \* **approxxy**(i, 0)) / **Zgang**(i)

**A**(i \* 2, 1) = (**R**(1, 0) \* f + **R**(1, 2) \* **approxxy**(i, 0)) / **Zgang**(i)

**A**(i \* 2, 2) = (**R**(2, 0) \* f + **R**(2, 2) \* **approxxy**(i, 0)) / **Zgang**(i)

**A**(i \* 2, 3) = (**approxxy**(i, 0) \* **approxxy**(i, 1)) \* **R**(1, 0) / f - (f + **Pow**(**approxxy**(i, 0), 2) / f) \* **R**(1, 1) - **approxxy**(i, 1) \* **R**(1, 2)

**A**(i \* 2, 4) = -**Pow**(**approxxy**(i, 0), 2) \* **Sin**(**Unknownum**(5)) / f - (**approxxy**(i, 0) \* **approxxy**(i, 1)) / f \* **Cos**(**Unknownum**(5)) - f \* **Sin**(**Unknownum**(5))

**A**(i \* 2, 5) = **approxxy**(i, 1)

**A**(i \* 2 + 1, 0) = (**R**(0, 1) \* f + **R**(0, 2) \* **approxxy**(i, 1)) / **Zgang**(i)

**A**(i \* 2 + 1, 1) = (**R**(1, 1) \* f + **R**(1, 2) \* **approxxy**(i, 1)) / **Zgang**(i)

**A**(i \* 2 + 1, 2) = (**R**(2, 1) \* f + **R**(2, 2) \* **approxxy**(i, 1)) / **Zgang**(i)

**A**(i \* 2 + 1, 3) = -(**approxxy**(i, 0) \* **approxxy**(i, 1)) \* **R**(1, 1) / f + (f + **Pow**(**approxxy**(i, 1), 2) / f) \* **R**(1, 0) - **approxxy**(i, 0) \* **R**(1, 2)

**A**(i \* 2 + 1, 4) = -**Pow**(**approxxy**(i, 1), 2) \* **Cos**(**Unknownum**(5)) / f - (**approxxy**(i, 0) \* **approxxy**(i, 1)) / f \* **Cos**(**Unknownum**(5)) - f \* **Sin**(**Unknownum**(5))

**A**(i \* 2 + 1, 5) = -**approxxy**(i, 0)

            Next

            Dim **L**(GCPNUMBER \* 2 - 1) As Double

            For i = 0 To GCPNUMBER \* 2 - 1

                If i Mod 2 = 0 Then

**L**(i) = **Xxy**(i \ 2, 0) - **approxxy**(i \ 2, 0)

                Else

**L**(i) = **Xxy**(i \ 2, 1) - **approxxy**(i \ 2, 1)   *'L*

                End If

            Next

            Dim **AT**(5, GCPNUMBER \* 2 - 1) As Double       *'AT*

            For i = 0 To 5

                For j = 0 To GCPNUMBER \* 2 - 1

**AT**(i, j) = **A**(j, i)

                Next

            Next

            Dim **Naa**(5, 5) As Double     *'NAA*

            For i = 0 To 5

                For j = 0 To 5

                    For k = 0 To GCPNUMBER \* 2 - 1

**Naa**(i, j) += **AT**(i, k) \* **A**(k, j)

                    Next

                Next

            Next

            Dim **NaaInv**(5, 5) As Double

            Dim inverse As Boolean = **GetMatrixInverse**(Naa, NN, NaaInv)    *'求NAA的逆*

            Dim **temp**(5) As Double

            For i = 0 To 5

                For j = 0 To GCPNUMBER \* 2 - 1

**temp**(i) += **AT**(i, j) \* **L**(j)

                Next

            Next

            For i = 0 To 5

**UnKnow**(i) = 0.0

                For j = 0 To 5

**UnKnow**(i) += **NaaInv**(i, j) \* **temp**(j)

                Next

            Next

            For i = 0 To 5

**Unknownum**(i) = **Unknownum**(i) + **UnKnow**(i)     *'误差方程*

            Next

            RepeatNumber += 1

        Loop While (**UnKnow**(3) >= 0.00002908882087 Or **UnKnow**(4) >= 0.00002908882087 Or **UnKnow**(5) >= 0.00002908882087) *'迭代*

    End Sub

    Function **GetMatrixInverse**(ByVal **src**(,) As Double, n As Integer, ByVal **des**(,) As Double) As Boolean

        Dim flag As Double = **GetA**(src, n)

        Dim **t**(NN - 1, NN - 1) As Double

        If 0 = flag Then

            Return False

        Else

**GetAStart**(src, n, t)

            For i = 0 To n - 1

                For j = 0 To n - 1

**des**(i, j) = **t**(i, j) / flag

                Next

            Next

        End If

        Return True

    End Function

    Function **GetA**(ByVal **arcs**(,) As Double, n As Integer) As Double

        If n = 1 Then

            Return **arcs**(0, 0)

        End If

        Dim ans As Double = 0

        Dim **temp**(NN - 1, NN - 1) As Double

        For i = 0 To n - 1

            For j = 0 To n - 2

                For k = 0 To n - 2

                    If k >= i Then

**temp**(j, k) = **arcs**(j + 1, k + 1)

                    Else **temp**(j, k) = **arcs**(j + 1, k)

                    End If

                Next

            Next

            Dim t As Double = **GetA**(temp, n - 1)

            If i Mod 2 = 0 Then

                ans += **arcs**(0, i) \* t

            Else

                ans -= **arcs**(0, i) \* t

            End If

        Next

        Return ans

    End Function

    Sub **GetAStart**(ByVal **arcs**(,) As Double, n As Integer, ByVal **ans**(,) As Double)

        If n = 1 Then

**ans**(0, 0) = 1

            Return

        End If

        Dim i%, j%, k%, t%

        Dim **temp**(NN - 1, NN - 1) As Double

        For i = 0 To n - 1

            For j = 0 To n - 1

                For k = 0 To n - 2

                    For t = 0 To n - 2

                        If k >= i Then

                            If t >= j Then

**temp**(k, t) = **arcs**(k + 1, t + 1)

                            Else

**temp**(k, t) = **arcs**(k + 1, t)

                            End If

                        Else

                            If t >= j Then

**temp**(k, t) = **arcs**(k, t + 1)

                            Else

**temp**(k, t) = **arcs**(k, t)

                            End If

                        End If

                    Next

                Next

**ans**(j, i) = **GetA**(temp, n - 1)

                If (i + j) Mod 2 = 1 Then

**ans**(j, i) = -**ans**(j, i)

                End If

            Next

        Next

    End Sub

End Module

Form1.vb

Imports System.Math

Public Class MainForm

    Dim PtA, PtB, PtC, PtD As ControlPoint *'控制点*

    Dim HelpString As String

    Dim **PutinDataStr**() As String

    Dim **PutDataStr**(24) As String

    Sub **ModeChange**()  *'输入模式改变事件*

        If RadioButton1.Checked = True Then

            Button1.Text = "开始导入"

            Label1.Text = "请按开始导入数据"

            GroupBox5.Text = "输入(导入文件)"

            ToolStripStatusLabel2.Text = "输入模式：导入文件模式。请单击开始导入按钮开始导入数据"

        Else

            Button1.Text = "开始输入"

            Label1.Text = "请按提示依次输入"

            GroupBox5.Text = "输入(手动输入)"

            ToolStripStatusLabel2.Text = "输入模式：手动输入模式。请单击开始输入按钮，按照提示开始输入数据"

        End If

        TextBox1.Enabled = False

        TextBox1.Visible = False

        TextBox2.Text = ""

        TextBox3.Text = ""

    End Sub

    Sub **OutputData**()    *'textbox2输出*

        TextBox2.Text = vbCrLf + "               ----- 输入数据表 -----" + vbCrLf + vbCrLf

        TextBox2.Text &= String.**Format**("摄影机主距 f：{0}", f) + vbCrLf + vbCrLf

        TextBox2.Text &= PtA.**PtInfo**() & PtB.**PtInfo**() & PtC.**PtInfo**() & PtD.**PtInfo**()

    End Sub

    Sub **AutoOutputData**()  *'导入文件的数据存放*

        Call **InputDataFile**(New OpenFileDialog)

        If DataStream = Nothing Then

            Exit Sub

        End If

        If DataStr.Length <> 25 Then

**MsgBox**("数据格式不匹配！" + vbCrLf + "请检查并重新组织数据。", 0 + 16 + 0, "错误提示")

            Exit Sub

        End If

        f = **Val**(DataStr$(0))

        PtA = New ControlPoint : PtB = New ControlPoint : PtC = New ControlPoint : PtD = New ControlPoint

        PtA.Name = **Val**(DataStr$(1)) : PtA.XCoorX = **Val**(DataStr$(2)) : PtA.XCoorY = **Val**(DataStr$(3)) : PtA.WCoorX = **Val**(DataStr$(4)) : PtA.WCoorY = **Val**(DataStr$(5)) : PtA.WCoorZ = **Val**(DataStr$(6))

        PtB.Name = **Val**(DataStr$(7)) : PtB.XCoorX = **Val**(DataStr$(8)) : PtB.XCoorY = **Val**(DataStr$(9)) : PtB.WCoorX = **Val**(DataStr$(10)) : PtB.WCoorY = **Val**(DataStr$(11)) : PtB.WCoorZ = **Val**(DataStr$(12))

        PtC.Name = **Val**(DataStr$(13)) : PtC.XCoorX = **Val**(DataStr$(14)) : PtC.XCoorY = **Val**(DataStr$(15)) : PtC.WCoorX = **Val**(DataStr$(16)) : PtC.WCoorY = **Val**(DataStr$(17)) : PtC.WCoorZ = **Val**(DataStr$(18))

        PtD.Name = **Val**(DataStr$(19)) : PtD.XCoorX = **Val**(DataStr$(20)) : PtD.XCoorY = **Val**(DataStr$(21)) : PtD.WCoorX = **Val**(DataStr$(22)) : PtD.WCoorY = **Val**(DataStr$(23)) : PtD.WCoorZ = **Val**(DataStr$(24))

        Call **OutputData**()  *'导入结果输出*

        Button1.Text = "开始计算"

        ToolStripStatusLabel2.Text = "导入文件完成。请检查导入数据表后，单击开始计算按钮开始计算"

    End Sub

    Sub **OutPut**()    *'计算结果输出*

        TextBox3.Text &= vbCrLf & String.**Format**("外方位元素:") & vbCrLf

        For i = 0 To 5

            TextBox3.Text &= **Unknownum**(i) & "   "

        Next

        TextBox3.Text &= vbCrLf & String.**Format**("精度评价方程") & vbCrLf

        For j = 0 To 5

            For i = 0 To 5

**UnKnown**(j, i) = **Sqrt**(**Abs**(**UnKnow**(j) \* **UnKnow**(i)) / (2 \* GCPNUMBER - 6))

                TextBox3.Text &= **UnKnown**(j, i) & "   "

            Next

            TextBox3.Text &= vbCrLf

        Next

        ToolStripStatusLabel2.Text = "计算完成，请保存结果"

        CmputRst = TextBox2.Text & vbCrLf & TextBox3.Text

    End Sub

    Private Sub **ToolStripButton3\_Click**(sender As Object, e As EventArgs) Handles ToolStripButton3.Click *'快捷栏开始计算事件*

        If Button1.Text = "开始计算" Then

            Call **Calculat**()

            Call **OutPut**()

        Else **MsgBox**("请输入数据后点击", vbOKOnly, "提示")

        End If

    End Sub

    Private Sub 导出文件ToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles 导出文件ToolStripMenuItem.Click *'菜单栏导出事件*

        If TextBox3.Text <> "" Then

**OutputRsutFile**(New SaveFileDialog)

        End If

    End Sub

    Private Sub 退出ToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles 退出ToolStripMenuItem.Click *'菜单栏退出事件*

        If **MsgBox**("你确认要退出程序吗？", MsgBoxStyle.OkCancel, "退出提示") = MsgBoxResult.Cancel Then

        Else End

        End If

    End Sub

    Private Sub 关于应用ToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles 关于应用ToolStripMenuItem.Click *'菜单栏关于事件*

**MsgBox**("《VB.NET测量程序设计》课程设计题目" & vbCrLf & "空间后方交会解影像方位元素" & vbCrLf & "2018011第11组作品" & vbCrLf & "组员：" & vbCrLf & "201801131常志鹏" & vbCrLf & "201801132黄义超" & vbCrLf & "201801133韩    洋" & vbCrLf & "201801134廉年刚", vbOKOnly, "关于应用")

    End Sub

    Private Sub 帮助ToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles 帮助ToolStripMenuItem.Click *'菜单栏帮助事件*

**MsgBox**("请根据左下角提示完成操作", vbOKOnly, "帮助")

    End Sub

    Private Sub **MainForm\_Load**(sender As Object, e As EventArgs) Handles Me.Load *'窗口初始化*

**ModeChange**()

    End Sub

    Private Sub **RadioButton1\_CheckedChanged**(sender As Object, e As EventArgs) Handles RadioButton1.CheckedChanged

**ModeChange**()

    End Sub

    Private Sub **RadioButton2\_CheckedChanged**(sender As Object, e As EventArgs) Handles RadioButton2.CheckedChanged

**ModeChange**()

    End Sub

    Private Sub **ToolStripButton1\_Click**(sender As Object, e As EventArgs) Handles ToolStripButton1.Click *'快捷栏导出文件事件*

        If TextBox3.Text <> "" Then

**OutputRsutFile**(New SaveFileDialog)

        End If

    End Sub

    Private Sub **ToolStripButton2\_Click**(sender As Object, e As EventArgs) Handles ToolStripButton2.Click *'快捷栏重新输入事件*

**ModeChange**()

    End Sub

    Private Sub 重新输入ToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles 重新输入ToolStripMenuItem.Click *'菜单栏重新输入事件*

**ModeChange**()

    End Sub

    Private Sub **Button1\_Click**(sender As Object, e As EventArgs) Handles Button1.Click

        Select Case Button1.Text

            Case Is = "开始输入" *'手动输入模式，开始输入*

                TextBox1.Enabled = True

                TextBox1.Visible = True

                Button1.Text = "确认"

                HelpString = "摄影机主距 f"

                Label1.Text = HelpString

                ToolStripStatusLabel2.Text = "请输入" & HelpString & ",并点击确认"

            Case Is = "开始导入"   *'文件导入模式，导入文件*

                Call **AutoOutputData**()

            Case Is = "开始计算"

                Call **Calculat**()   *'文件导入模式，计算外方元素*

                Call **OutPut**()

            Case Is = "确认"

                If TextBox1.Text <> "" Then

                    Select Case Label1.Text

                        Case "摄影机主距 f"

                            f = **Val**(TextBox1.Text) *'手动输入f*

                            TextBox2.Text = vbCrLf + "               ----- 输入数据表 -----" + vbCrLf + vbCrLf

                            TextBox2.Text &= String.**Format**("摄影机主距 f：{0}", f) + vbCrLf + vbCrLf

                            TextBox1.Text = ""

                            HelpString = "控制点1信息"

                            Label1.Text = HelpString

                            ToolStripStatusLabel2.Text = "请输入" & HelpString & ",并点击确认。" & "格式：控制点号,像点x,像点y,物点X,物点Y,物点Z"

**PutDataStr**(0) = f

                        Case "控制点1信息"

                            PutinDataStr = TextBox1.Text.**Split**(",")

                            PtA = New ControlPoint  *'手动输入控制点1信息*

                            PtA.Name = **Val**(PutinDataStr$(0)) : PtA.XCoorX = **Val**(PutinDataStr$(1)) : PtA.XCoorY = **Val**(PutinDataStr$(2)) : PtA.WCoorX = **Val**(PutinDataStr$(3)) : PtA.WCoorY = **Val**(PutinDataStr$(4)) : PtA.WCoorZ = **Val**(PutinDataStr$(5))

                            For i = 0 To 5

**PutDataStr**(i + 1) = **PutinDataStr**(i)

                            Next

                            TextBox2.Text &= PtA.**PtInfo**()

                            TextBox1.Text = ""

                            HelpString = "控制点2信息"

                            Label1.Text = HelpString

                            ToolStripStatusLabel2.Text = "请输入" & HelpString & ",并点击确认。" & "格式：控制点号,像点x,像点y,物点X,物点Y,物点Z"

                        Case "控制点2信息"

                            PutinDataStr = TextBox1.Text.**Split**(",") *'手动输入控制点2信息*

                            PtB = New ControlPoint

                            PtB.Name = **Val**(PutinDataStr$(0)) : PtB.XCoorX = **Val**(PutinDataStr$(1)) : PtB.XCoorY = **Val**(PutinDataStr$(2)) : PtB.WCoorX = **Val**(PutinDataStr$(3)) : PtB.WCoorY = **Val**(PutinDataStr$(4)) : PtB.WCoorZ = **Val**(PutinDataStr$(5))

                            For i = 0 To 5

**PutDataStr**(i + 7) = **PutinDataStr**(i)

                            Next

                            TextBox2.Text &= PtB.**PtInfo**()

                            TextBox1.Text = ""

                            HelpString = "控制点3信息"

                            Label1.Text = HelpString

                            ToolStripStatusLabel2.Text = "请输入" & HelpString & ",并点击确认。" & "格式：控制点号,像点x,像点y,物点X,物点Y,物点Z"

                        Case "控制点3信息"

                            PutinDataStr = TextBox1.Text.**Split**(",") *'手动输入控制点3信息*

                            PtC = New ControlPoint

                            PtC.Name = **Val**(PutinDataStr$(0)) : PtC.XCoorX = **Val**(PutinDataStr$(1)) : PtC.XCoorY = **Val**(PutinDataStr$(2)) : PtC.WCoorX = **Val**(PutinDataStr$(3)) : PtC.WCoorY = **Val**(PutinDataStr$(4)) : PtC.WCoorZ = **Val**(PutinDataStr$(5))

                            For i = 0 To 5

**PutDataStr**(i + 13) = **PutinDataStr**(i)

                            Next

                            TextBox2.Text &= PtC.**PtInfo**()

                            TextBox1.Text = ""

                            HelpString = "控制点4信息"

                            Label1.Text = HelpString

                        Case "控制点4信息"

                            PutinDataStr = TextBox1.Text.**Split**(",") *'手动输入控制点4信息*

                            PtD = New ControlPoint

                            PtD.Name = **Val**(PutinDataStr$(0)) : PtD.XCoorX = **Val**(PutinDataStr$(1)) : PtD.XCoorY = **Val**(PutinDataStr$(2)) : PtD.WCoorX = **Val**(PutinDataStr$(3)) : PtD.WCoorY = **Val**(PutinDataStr$(4)) : PtD.WCoorZ = **Val**(PutinDataStr$(5))

                            For i = 0 To 5

**PutDataStr**(i + 19) = **PutinDataStr**(i)

                            Next

                            TextBox2.Text &= PtD.**PtInfo**()

                            TextBox1.Text = ""

                            TextBox1.Enabled = False

                            TextBox1.Visible = False

                            HelpString = "输入完成"

                            Label1.Text = HelpString

                            Button1.Text = "开始计算"

                            DataStr = PutDataStr

                    End Select

                End If

        End Select

    End Sub

End Class

1. **结束语**

SpaceResection目前只是一个连软件都称不上的程序，并且通过这个文档也只能够了解它的很小一部分，所以如果你要详细的了解它，只是通过这个文档

是绝对不够的。进一步的阅读就只有源代码了，因为时间的关系我们也来不及详细地写出系统的设计

如果你在使用了SpaceResection以后有关于它的建议或者想法，无论赞扬的还是批评的或者其它方面的，欢迎反馈给我们

感谢你的支持