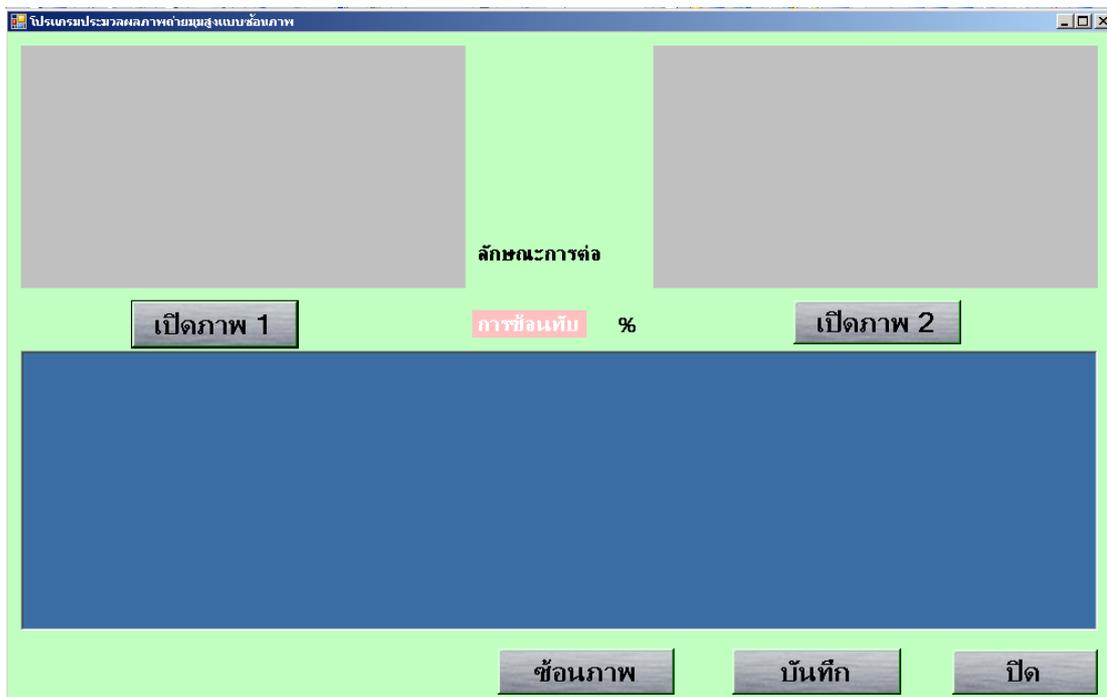


ภาคผนวก ก
ต้นฉบับโปรแกรม

โค้ดโปรแกรมประมวลผลภาพถ่ายมุมสูงแบบซ้อนภาพ



```
Imports System.IO
```

```
Imports System.Drawing.Bitmap
```

```
Public Class Form1
```

ปุ่มเปิดภาพ 1

```
Private Sub upld_btn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles upld_btn.Click
```

```
Dim open_file As New OpenFileDialog
```

```
open_file.Filter = "Pictures Files| *.bmp;*.jpg;*.gif;*.wmf;*.tif;*.png"
```

```
Try
```

```
If open_file.ShowDialog() = DialogResult.OK Then
```

```
picture_bx1.Image = Image.FromFile(open_file.FileName)
```

```
Else
```

```
MsgBox("Please select a picture", vbInformation, "Pictures Files")
```

```
End If
```

```

Catch ex As Exception
    MsgBox(ex.Message, vbCritical, "Pictures Files")
End Try
End Sub

```

ปุ่มเปิดภาพ 2

```

Private Sub upld_btn2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles upld_btn2.Click
    Dim open_file2 As New OpenFileDialog
    open_file2.Filter = "Pictures Files| *.bmp;*.jpg;*.gif;*.wmf;*.tif;*.png"
    Try
        If open_file2.ShowDialog() = DialogResult.OK Then
            picture_bx2.Image = Image.FromFile(open_file2.FileName)
        Else
            MsgBox("Please select a picture", vbInformation, "Pictures Files")
        End If
    Catch ex As Exception
        MsgBox(ex.Message, vbCritical, "Pictures Files")
    End Try
End Sub

```

ปุ่มซ่อนภาพ

```

Private Sub bt_1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles bt_1.Click
    Dim Image_1 As New Bitmap(picture_bx1.Image)
    Dim Px1 As Integer
    Dim Py1 As Integer
    Dim strL1 As String = ""
    Dim Image_2 As New Bitmap(picture_bx2.Image)
    Dim Px2 As Integer

```

```

Dim Py2 As Integer
Dim strL2 As String = ""
Dim Image_1_Pixel(Image_1.Width - 1, Image_1.Height - 1) As Color
Dim Image_2_Pixel(Image_2.Width - 1, Image_2.Height - 1) As Color
Dim DD As String = ""
For Px1 = Image_1.Width - 1 To 0 Step -1
    For Py1 = Image_1.Height - 1 To 0 Step -1
        Image_1_Pixel(Px1, Py1) = Image_1.GetPixel(Px1, Py1)
    Next
Next

For Px2 = Image_2.Width - 1 To 0 Step -1
    For Py2 = Image_2.Height - 1 To 0 Step -1
        Image_2_Pixel(Px2, Py2) = Image_2.GetPixel(Px2, Py2)
    Next
Next

Dim OverlapValue_1 As Integer
OverlapValue_1 = CheckOverlap_1(Image_1_Pixel, Image_2_Pixel, Image_1, Image_2)
Dim OverlapValue_2 As Integer
OverlapValue_2 = CheckOverlap_2(Image_1_Pixel, Image_2_Pixel, Image_1, Image_2)

Dim OverlapValue_3 As Integer
OverlapValue_3 = CheckOverlap_3(Image_1_Pixel, Image_2_Pixel, Image_1, Image_2)
Dim OverlapValue_4 As Integer
OverlapValue_4 = CheckOverlap_4(Image_1_Pixel, Image_2_Pixel, Image_1, Image_2)

If OverlapValue_1 > OverlapValue_2 And OverlapValue_1 > OverlapValue_3 And
OverlapValue_1 > OverlapValue_4 And OverlapValue_1 <> OverlapValue_2 Then
End Sub

```

```

Private Function CheckOverlap_1(ByVal Image_1_Pixel(.) As Color, ByVal Image_2_Pixel(.)
As Color, ByVal Image_1 As Bitmap, ByVal Image_2 As Bitmap) As Integer
    Dim overlap As Integer = 0
    Dim px, py, px2, nj As Integer
    nj = 0
    px = Image_1.Width - 1
    px2 = Image_2.Height - 1
    Dim njj, nii As Integer
    njj = Image_2.Height - 1
    nii = Image_1.Height - 1
    px = Image_1.Width - 1
    py = Image_2.Width - 1
    Dim overlap_array(Image_1.Height - 1) As Integer
    Dim ARy_Mode(Image_1.Width - 1) As Integer
    Dim Arry_n2(126) As Integer
    Dim t As Integer

    For n2 = 0 To 126 Step 1
        t = n2
        nj = 0
        While (nj <= Image_1.Height - 1)
            px = Image_1.Width - 1
            py = Image_2.Width - 1
            overlap = 0
            While (px >= 0)
                While (py >= 0)
                    If ComparePixel(Image_1_Pixel(px, nj), Image_2_Pixel(py, nj), t) Then
                        While ComparePixel(Image_1_Pixel(px, nj), Image_2_Pixel(py, nj), t) And (py
>= 0)

```

```

        If py > 0 Then
            overlap = overlap + 1
            px = px - 1
            py = py - 1

        ElseIf py = 0 Then
            overlap = overlap + 1
            Exit While
        End If
    End While
End While
py = py - 1
End While
px = px - 1
End While
overlap_array(nj) = overlap
nj = nj + 1
End While

Dim myMin As Integer
Dim n As Integer

Dim AvgOverlap As Integer = 0
Dim SumOverlap As Integer = 0

myMin = Integer.MaxValue
For n = 0 To overlap_array.Length - 1
    SumOverlap = SumOverlap + overlap_array(n)
    If (overlap_array(n) < myMin) And overlap_array(n) > 0 Then
        myMin = overlap_array(n)
    End If

```

```

Next
AvgOverlap = SumOverlap / overlap_array.Length
ARRy_Mode(myMin) = ARRy_Mode(myMin) + 1
Arry_n2(n2) = myMin
Next

Dim mxIdx As Integer = 0
For i = 0 To ARRy_Mode.Length - 1
    If i >= 40 Then
    End If
    If ARRy_Mode(mxIdx) < ARRy_Mode(i) Then
        mxIdx = i
    End If
Next
Return mxIdx
End Function

Private Function CheckOverlap_2(ByVal Image_1_Pixel(,) As Color, ByVal Image_2_Pixel(,)
As Color, ByVal Image_1 As Bitmap, ByVal Image_2 As Bitmap) As Integer
    Dim overlap_4 As Integer = 0
    Dim px, py, nj As Integer
    nj = 0
    px = 0
    py = 0

    Dim overlap_array(Image_1.Width - 1) As Integer
    Dim ARRy_Mode(Image_1.Width - 1) As Integer
    Dim t As Integer

    For n2 = 0 To 126 Step 1

```

```

t = n2
nj = 0
While (nj <= Image_1.Height - 1)
  px = 0
  py = 0
  overlap_4 = 0
  While (px <= Image_1.Width - 1)
    While (py <= Image_2.Width - 1)
      If ComparePixel(Image_1_Pixel(px, nj), Image_2_Pixel(py, nj), t) Then
        While ComparePixel(Image_1_Pixel(px, nj), Image_2_Pixel(py, nj), t) And (py
< Image_1.Width - 1)
          If py < Image_1.Width - 1 Then
            overlap_4 = overlap_4 + 1
            px = px + 1
            py = py + 1

          ElseIf py = Image_1.Width - 1 Then
            overlap_4 = overlap_4 + 1
            Exit While
          End If
        End While
      End If
      py = py + 1
    End While
    px = px + 1
  End While
  overlap_array(nj) = overlap_4
  nj = nj + 1
End While

```

```
Dim myMin As Integer
```

```
Dim n As Integer
```

```
Dim AvgOverlap As Integer = 0
```

```
Dim SumOverlap As Integer = 0
```

```
myMin = Integer.MaxValue
```

```
For n = 0 To overlap_arr.Length - 1
```

```
    SumOverlap = SumOverlap + overlap_arr(n)
```

```
    If (overlap_arr(n) < myMin) And overlap_arr(n) > 0 Then
```

```
        myMin = overlap_arr(n)
```

```
    End If
```

```
Next
```

```
AvgOverlap = SumOverlap / overlap_arr.Length
```

```
ARRy_Mode(myMin) = ARRy_Mode(myMin) + 1
```

```
Next
```

```
Dim mxIdx As Integer = 0
```

```
For i = 0 To ARRy_Mode.Length - 1
```

```
    If ARRy_Mode(mxIdx) < ARRy_Mode(i) Then
```

```
        mxIdx = i
```

```
    End If
```

```
Next
```

```
Return mxIdx
```

```
End Function
```

```
Private Function CheckOverlap_3(ByVal Image_1_Pixel(,) As Color, ByVal Image_2_Pixel(,)
As Color, ByVal Image_1 As Bitmap, ByVal Image_2 As Bitmap) As Integer
```

```
    Dim overlap_4 As Integer = 0
```

```
    Dim px, py, nj As Integer
```

```

Dim overlap_array(Image_1.Width - 1) As Integer
Dim ARRY_Mode(Image_1.Width - 1) As Integer
Dim t As Integer

For n2 = 0 To 126 Step 1
    t = n2
    nj = 0
    While (nj <= Image_1.Width - 1)
        px = 0
        py = 0
        overlap_4 = 0
        While (px <= Image_1.Height - 1)
            While (py <= Image_2.Height - 1)
                If ComparePixel(Image_1_Pixel(nj, px), Image_2_Pixel(nj, py), t) Then
                    While ComparePixel(Image_1_Pixel(nj, px), Image_2_Pixel(nj, py), t) And (py
<= Image_2.Height - 1)
                        If py < Image_2.Height - 1 Then
                            overlap_4 = overlap_4 + 1
                            px = px + 1
                            py = py + 1
                        ElseIf py = Image_2.Height - 1 Then
                            overlap_4 = overlap_4 + 1
                            Exit While
                        End If
                    End While
                End If
                py = py + 1
            End While
            px = px + 1
        End While
    End While

```

```

    End While
    overlap_arry(nj) = overlap_4
    nj = nj + 1
End While
Dim myMin As Integer
Dim n As Integer

Dim AvgOverlap As Integer = 0
Dim SumOverlap As Integer = 0

myMin = Integer.MaxValue
For n = 0 To overlap_arry.Length - 1
    SumOverlap = SumOverlap + overlap_arry(n)
    If (overlap_arry(n) < myMin) And overlap_arry(n) > 0 Then
        myMin = overlap_arry(n)
    End If
Next
ARRy_Mode(myMin) = ARRy_Mode(myMin) + 1
AvgOverlap = SumOverlap / overlap_arry.Length
ARRy_Mode(myMin) = ARRy_Mode(myMin) + 1

Next

Dim mxIdx As Integer = 0
For i = 0 To ARRy_Mode.Length - 1
    If ARRy_Mode(mxIdx) < ARRy_Mode(i) Then
        mxIdx = i
    End If
Next

Return mxIdx

```

End Function

```

Private Function CheckOverlap_4(ByVal Image_1_Pixel(.) As Color, ByVal Image_2_Pixel(.)
As Color, ByVal Image_1 As Bitmap, ByVal Image_2 As Bitmap) As Integer
    Dim overlap_4 As Integer = 0
    Dim px, py, nj As Integer
    nj = 0
    px = Image_1.Width - 1
    py = Image_2.Width - 1
    Dim overlap_array(Image_1.Width - 1) As Integer
    Dim ARRAY_Mode(Image_1.Width - 1) As Integer
    Dim t As Integer

    For n2 = 0 To 126 Step 1
        t = n2
        nj = 0
        While (nj <= Image_1.Width - 1)
            px = Image_1.Height - 1
            py = Image_2.Height - 1
            overlap_4 = 0
            While (px >= 0)
                While (py >= 0)
                    If ComparePixel(Image_1_Pixel(nj, px), Image_2_Pixel(nj, py), t) Then
                        While ComparePixel(Image_1_Pixel(nj, px), Image_2_Pixel(nj, py), t) And (py
>= 0)
                            If py > 0 Then
                                overlap_4 = overlap_4 + 1
                                px = px - 1
                                py = py - 1

```

```

        ElseIf py = 0 Then
            overlap_4 = overlap_4 + 1
            Exit While
        End If
    End While

    End If

    py = py - 1
End While

px = px - 1
End While

overlap_array(nj) = overlap_4

    nj = nj + 1
End While

Dim myMin As Integer
Dim n As Integer

Dim AvgOverlap As Integer = 0
Dim SumOverlap As Integer = 0

myMin = Integer.MaxValue
For n = 0 To overlap_array.Length - 1
    SumOverlap = SumOverlap + overlap_array(n)
    If (overlap_array(n) < myMin) And overlap_array(n) > 0 Then
        myMin = overlap_array(n)
    End If
Next

AvgOverlap = SumOverlap / overlap_array.Length
ARRy_Mode(myMin) = ARRy_Mode(myMin) + 1

```

```

Next
Dim mxIdx As Integer = 0
For i = 0 To ARRY_Mode.Length - 1
    If ARRY_Mode(mxIdx) < ARRY_Mode(i) Then
        mxIdx = i
    End If
Next
Return mxIdx
End Function

Private Function ComparePixel(ByVal Image_1_Pixel As Color, ByVal Image_2_Pixel As
Color, ByVal mau As Integer) As Boolean
    Dim res As Boolean = False
    Dim k As Integer
    k = mau

    If (Image_1_Pixel.R >= (Image_2_Pixel.R - k) And Image_1_Pixel.R <= (Image_2_Pixel.R
+ k)) Or (Image_1_Pixel.G >= (Image_2_Pixel.G - k) And Image_1_Pixel.G <=
(Image_2_Pixel.G + k)) Or (Image_1_Pixel.B >= (Image_2_Pixel.B - k) And Image_1_Pixel.B
<= (Image_2_Pixel.B + k)) Then
res = True
    End If
    Return res
End Function
End Sub

```

ปุ่มบันทึก

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    SaveFileDialog1.FileName = OpenFileDialog1.SafeFileName

```

```
SaveFileDialog1.Filter = ("ไฟล์รูปภาพ JPEG,GIF,PNG,BMP|*.jpg; *.gif; *.png; *.bmp")
If SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
    PictureBox1.Image.Save(SaveFileDialog1.FileName)
    MessageBox.Show("Save complete ", "เปิดรูปภาพ", MessageBoxButtons.OK,
    MessageBoxIcon.Information)
End If
End Sub
```

ปุ่มปิด

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
    Me.Close()
End Sub
End Class
```