* Amb les coordenades obtingudes es calcula el pla promig i s’obte el vector normal al pla i l’index de colinearitat i colplanarietat com a indicadors de lo bondat i projectada a l’stereoplot

Sub calculparametersfeatures()

seleccio = -1

trazelengthtemporal = 0

For i = contadorz To valortemporal2 Step -1 'vigilar el 0

If contadorxyz(i) <> 0 Then

seleccio = seleccio + 1

ReDim Preserve x(seleccio)

ReDim Preserve y(seleccio)

ReDim Preserve z(seleccio)

x(seleccio) = solx(i)

y(seleccio) = soly(i)

z(seleccio) = solz(i)

If i <> 0 Then

If contadorxyz(i - 1) <> 0 Then

trazelengthtemporal += Math.Sqrt((Math.Pow((solx(i) - solx(i - 1)), 2)) + (Math.Pow(soly(i) - soly(i - 1), 2)) + (Math.Pow(solz(i) - solz(i - 1), 2)))

End If

End If

Else

i = 0

End If

Next

calculvectors()

calculdips()

Dim Mtemporal As Double

Dim Ktemporal As Double

Dim dipdirtemporal As Double

Dim diptemporal As Double

Dim popul As Double

'ReDim Preserve trazelength(contadorobjecte)

Mtemporal = prop3

Ktemporal = kapa

popul = seleccio + 1

diptemporal = pendent

dipdirtemporal = orientacio

If EstatForm4 = False Then

Form1.Label13.Text = Format(dipdirtemporal, "#.##")

Form1.Label14.Text = Format(diptemporal, "#.##")

If Mtemporal < 4 Then

Form1.Label15.BackColor = Color.Red

Else

Form1.Label15.BackColor = Color.FromKnownColor(KnownColor.Control)

End If

Form1.Label15.Text = Format(Mtemporal, "#.##")

If Ktemporal > 0.8 Then

Form1.Label16.BackColor = Color.Red

Else

Form1.Label16.BackColor = Color.FromKnownColor(KnownColor.Control)

End If

Form1.Label16.Text = Format(Ktemporal, "#.##")

Form1.Label17.Text = popul

Form1.Label18.Text = Format(trazelengthtemporal, "#.##")

Form1.Label21.Text = "Polyline " + Str(contadorobjecte + 1)

Calculatepos()

Dim nx As Double

Dim ny As Double

Dim points As Point

nx = 1 \* ((Math.Cos((90 - diptemporal) \* (Math.PI / 180)) \* Math.Sin((dipdirtemporal - 180) \* (Math.PI / 180))) \* Math.Sqrt(1 / (1 - (-(Math.Sin((90 - diptemporal) \* (Math.PI / 180)))))))

ny = -1 \* ((Math.Cos((90 - diptemporal) \* (Math.PI / 180)) \* Math.Cos((dipdirtemporal - 180) \* (Math.PI / 180))) \* Math.Sqrt(1 / (1 - (-(Math.Sin((90 - diptemporal) \* (Math.PI / 180)))))))

points.X = 100 \* nx + 20 + 100 - 2

points.Y = 100 \* ny + 240 + 100 - 2

Form1.CreateGraphics.FillEllipse(Brushes.Black, points.X - 1, points.Y - 1, 6, 6)

End If

End Sub

Function calculdips() As Double

If vector\_i = 0 And vector\_j = 0 Then '\*\*\*la traça és horitzontal\*\*\*

orientacio = 0

pendent = 0

GoTo line200

End If

'\*\*\*calculem dip i dip direction\*\*\*

pendent = Math.Atan(Math.Sqrt((vector\_i \* vector\_i) + (vector\_j \* vector\_j)) / vector\_k)

pendent = (pendent \* 180) / Math.PI

If vector\_j <> 0 Then

orientacio = Math.Atan(vector\_i / vector\_j)

orientacio = (orientacio \* 180) / Math.PI

Else

If vector\_i < 0 Then '\*\*\*si m=0, llavors l<>0 perquè ja hem eliminat les traces horitzontals\*\*\*

orientacio = -90

Else

orientacio = 90

End If

End If

If vector\_j < 0 Then

orientacio = 180 + orientacio

Else

If vector\_i < 0 Then

orientacio = 360 + orientacio

Else

End If

End If

line200:

'az1 = CInt(a1)

Return Math.Abs(CInt(pendent))

End Function

Sub calculvectors()

' Dim i As Integer, seleccio As Integer

Dim Vec1 As Double, Vec2 As Double, Vec3 As Double

' Dim xmean As Double, ymean As Double, zmean As Double

Dim sumx As Double, sumy As Double, sumz As Double

Dim sumyz As Double, sumxz As Double, sumxy As Double

Dim sumxsq As Double, sumysq As Double, sumzsq As Double

Dim summodpt As Double

Dim modpt As Double

Dim c As Double

'\*\*\*\*\*\*agafem els punts escollits\*\*\*\*\*\*\*

If contadorz > 5 Then

sumx = 0 : sumy = 0 : sumz = 0

'\*\*\*calculem sumatoris\*\*\*

sumyz = 0 : sumxz = 0 : sumxy = 0

sumxsq = 0 : sumysq = 0 : sumzsq = 0

summodpt = 0

'\*\*\*calculem sumatoris\*\*\*

For i = 0 To seleccio

sumyz = sumyz + ((y(i)) - y.Average) \* ((z(i)) - z.Average)

sumxz = sumxz + ((x(i)) - x.Average) \* ((z(i)) - z.Average)

sumxy = sumxy + ((x(i)) - x.Average) \* ((y(i)) - y.Average)

sumxsq = sumxsq + Math.Pow(((x(i)) - x.Average), 2)

sumysq = sumysq + Math.Pow(((y(i)) - y.Average), 2)

sumzsq = sumzsq + Math.Pow(((z(i)) - z.Average), 2)

'\*\*\*aquest es pels eigenvalues\*\*\*

'\*\*\*calculo l'arrel del modul xq vull el sumatori dls quadrats dls moduls\*\*\*

modpt = Math.Pow((x(i) - x.Average), 2) + Math.Pow((y(i) - y.Average), 2) + Math.Pow((z(i) - z.Average), 2)

summodpt = summodpt + modpt

Next i

'\*\*\*Ara ve el càlcul\*\*\*

Dim n As Integer = 3

Dim a(3, 3) As Double

Dim d(3) As Double

Dim V(3, 3) As Double

Dim b(3) As Double

Dim zz(3) As Double

Dim sm As Double, g As Double, h As Double, s As Double, t As Double, p As Double

'redim c As Double

Dim tau As Double, tresh As Double, theta As Double

Dim Nrot As Integer

'redim i As Integer

Dim j As Integer, k As Integer

Dim ip As Integer, iq As Integer

Dim prop4 As String

a(1, 1) = sumxsq

a(2, 2) = sumysq

a(3, 3) = sumzsq

a(1, 2) = sumxy

a(2, 1) = sumxy

a(1, 3) = sumxz

a(3, 1) = sumxz

a(2, 3) = sumyz

a(3, 2) = sumyz

For ip = 1 To n

For iq = 1 To n

V(ip, iq) = 0

Next iq

V(ip, ip) = 1

Next ip

For ip = 1 To n

b(ip) = a(ip, ip)

d(ip) = b(ip)

zz(ip) = 0

Next ip

Nrot = 0

For i = 1 To 50

sm = 0

For ip = 1 To (n - 1)

For iq = (ip + 1) To n

sm = sm + Math.Abs(a(ip, iq))

Next iq

Next ip

If sm = 0 Then GoTo LINE123

If i < 4 Then

tresh = 0.2 \* sm / n ^ 2

Else

tresh = 0

End If

For ip = 1 To (n - 1)

For iq = (ip + 1) To n

g = 100 \* Math.Abs(a(ip, iq))

If i > 4 And (Math.Abs(d(ip)) + g) = Math.Abs(d(ip)) And (Math.Abs(d(iq)) + g) = Math.Abs(d(iq)) Then

a(ip, iq) = 0

ElseIf Math.Abs(a(ip, iq)) > tresh Then

h = d(iq) - d(ip)

If (Math.Abs(h) + g) = Math.Abs(h) Then

t = a(ip, iq) / h

Else

theta = 0.5 \* h / a(ip, iq)

t = 1 / (Math.Abs(theta) + Math.Sqrt(1 + theta ^ 2))

If theta < 0 Then t = -t

End If

c = 1 / Math.Sqrt(1 + Math.Pow(t, 2))

s = t \* c

tau = s / (1 + c)

h = t \* a(ip, iq)

zz(ip) = zz(ip) - h

zz(iq) = zz(iq) + h

d(ip) = d(ip) - h

d(iq) = d(iq) + h

a(ip, iq) = 0

For j = 1 To (ip - 1)

g = a(j, ip)

h = a(j, iq)

a(j, ip) = g - s \* (h + g \* tau)

a(j, iq) = h + s \* (g - h \* tau)

Next j

For j = (ip + 1) To (iq - 1)

g = a(ip, j)

h = a(j, iq)

a(ip, j) = g - s \* (h + g \* tau)

a(j, iq) = h + s \* (g - h \* tau)

Next j

For j = (iq + 1) To n

g = a(ip, j)

h = a(iq, j)

a(ip, j) = g - s \* (h + g \* tau)

a(iq, j) = h + s \* (g - h \* tau)

Next j

For j = 1 To n

g = V(j, ip)

h = V(j, iq)

V(j, ip) = g - s \* (h + g \* tau)

V(j, iq) = h + s \* (g - h \* tau)

Next j

Nrot = Nrot + 1

End If

Next iq

Next ip

For ip = 1 To n

b(ip) = b(ip) + zz(ip)

d(ip) = b(ip)

zz(ip) = 0

Next ip

Next i

LINE123:

'\*\*\*ara ordenem\*\*\*

For i = 1 To n - 1

k = i

p = d(i)

For j = i + 1 To n

If d(j) > p Then

k = j

p = d(j)

End If

Next j

If k <> i Then

d(k) = d(i)

d(i) = p

For j = 1 To n

p = V(j, i)

V(j, i) = V(j, k)

V(j, k) = p

Next j

End If

Next i

'\*\*\*parem si la matriu d'eigenvectors és la identitat (per exemple si mesurem cabussament d'isolínies)\*\*\*

'If V(1, 1) = 0 And V(2, 2) = 0 And V(3, 3) = 0 Then

'ret = MbeMessageBox("Solution not possible with current eigenvector algorithm", 2048)

' Exit Sub

'End If

'Dim ll As Double, mm As Double, nn As Double

'Dim B1 As Double, a1 As Double

'\*\*\*volem el vector 3 (criteri Woodcock (1977))\*\*\*

'\*\*\*redrecem el vector per a què pugi\*\*\*

Dim prop1 As Double, prop2 As Double

If V(3, 3) < 0 Then

vector\_i = -V(1, 3)

vector\_j = -V(2, 3)

vector\_k = -V(3, 3)

Else

vector\_i = V(1, 3)

vector\_j = V(2, 3)

vector\_k = V(3, 3)

End If

'afegit

If V(3, 3) < 0 Then

Vec1 = -V(1, 1)

Vec2 = -V(2, 1)

Vec3 = -V(3, 1)

Else

Vec1 = V(1, 1)

Vec2 = V(2, 1)

Vec3 = V(3, 1)

End If

'afegit

prop1 = Math.Log(d(1) / d(2))

prop2 = Math.Log(d(2) / d(3))

prop3 = Math.Log(d(1) / d(3)) 'Aixo es la M

' Form1.Label1.Text = Format(prop3, "#.###")

If vector\_i = 0 And vector\_j = 1 And vector\_k = 0 Then prop3 = 10

If vector\_i = 0 And vector\_j = 0 And vector\_k = 1 Then prop3 = 10

If vector\_i = 1 And vector\_j = 0 And vector\_k = 0 Then prop3 = 10

prop4 = Str(prop3)

If prop4 = "Infinito" Then prop3 = 1035

kapa = prop1 / prop2 ' Això es la K

Else

vector\_i = 0

vector\_j = 0

vector\_k = 0

End If

If chivato1 = False Then

ReDim Preserve M(contadorobjecte)

ReDim Preserve K(contadorobjecte)

ReDim Preserve DipDir(contadorobjecte)

ReDim Preserve Dip(contadorobjecte)

ReDim Preserve popul(contadorobjecte)

ReDim Preserve trazelength(contadorobjecte)

ReDim Preserve centroidex(contadorobjecte)

ReDim Preserve centroidey(contadorobjecte)

ReDim Preserve centroidez(contadorobjecte)

ReDim Preserve veci1(contadorobjecte)

ReDim Preserve vecj1(contadorobjecte)

ReDim Preserve veck1(contadorobjecte)

veci1(conti) = vector\_i

vecj1(conti) = vector\_j

veck1(conti) = vector\_k

M(conti) = prop3

K(conti) = kapa

popul(conti) = seleccio + 1

calculdips()

Dip(conti) = pendent

DipDir(conti) = orientacio

centroidex(conti) = x.Average

centroidey(conti) = y.Average

centroidez(conti) = z.Average

Else

If EstatForm4 = True Then

ReDim Preserve M4(newitem)

ReDim Preserve K4(newitem)

ReDim Preserve DipDir4(newitem)

ReDim Preserve Dip4(newitem)

ReDim Preserve popul4(newitem)

ReDim Preserve trazelength4(newitem)

ReDim Preserve centroidex4(newitem)

ReDim Preserve centroidey4(newitem)

ReDim Preserve centroidez4(newitem)

M4(newitem) = prop3

K4(newitem) = kapa

popul4(newitem) = x.Count - 1

Dip4(newitem) = pendent

DipDir4(newitem) = orientacio

centroidex4(newitem) = x.Average

centroidey4(newitem) = y.Average

centroidez4(newitem) = z.Average

Form4.Label28.Text = Format(centroidex4(newitem), "#.##")

Form4.Label27.Text = Format(centroidey4(newitem), "#.##")

Form4.Label26.Text = Format(centroidez4(newitem), "#.##")

Form4.Label15.Text = Format(M4(newitem), "#.##")

Form4.Label16.Text = Format(K4(newitem), "#.##")

Form4.Label23.Text = popul4(newitem)

Form4.Label14.Text = Format(Dip4(newitem), "#.##")

Form4.Label13.Text = Format(DipDir4(newitem), "#.##")

End If

End If

End Sub