

---

Subject: can you translate IDL5.1 to IDL2.x...(Hough transform source)

Posted by [Duck-Jin Park](#) on Sun, 27 Jun 1999 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Can you translate following source, IDL version 5.1 to IDL version 2.x  
following source is hough transform...

i want to independently execution file on IDL version 2.x. Because I  
have not idl version 5.1.

Please help me...

\*\*\*\*\* \*\*

'AUTO' : begin

WIDGET\_CONTROL,info.sourcewin,GET\_UValue=SOURCE\_IMAGE  
LOADCT,39

w=[[-1,-1,-1],[-1,8,-1],[-1,-1,-1]] & s=1

C=SOBEL(SOURCE\_IMAGE);CONVOL(SOURCE\_IMAGE,W,S)

SIZ=SIZE(SOURCE\_IMAGE) & N=SIZ(1) & M=SIZ(2)

posi=where(c ne 0)

XPTS=POSI MOD N

YPTS=POSI / N

theta=INDGEN(51)/50\*!PI-!pi/2

; Set up a vector in degrees for plotting

degrees=theta\*180/!pi

; Set up arrays of cosine and sine values

cosv=cos(theta)

sinv=sin(theta)

WIDGET\_CONTROL,info.drawTE,GET\_VALUE=tem

WSET,tem & erase

PLOT,degrees,xpts[0]\*cosv+ypts[0]\*sinv,xrange=[-90,90],\$

YRANGE=[-SQRT(N^2+M^2),SQRT(N^2+M^2)],xticklen=1,yticklen=1, \$

xticks=4,xtickv=[-90,-45,0,45,90],/nodata,\$

TITLE='Parameter Space (Polar coordinate)',

Xtitle='Angle',Ytitle='Ro'

kpts=MIN([N\_ELEMENTS(xpts),N\_ELEMENTS(ypts)])

; Plot the parametric curves for each data point

FOR k=0,kpts-1 DO \$

OPLOT,degrees,xpts[k]\*cosv+ypts[k]\*sinv,color=FIX(k/FLOAT(KPTS)\*255)

QUAN\_ang=2 & QUAN\_RO=2; & THRESHOLD=30

BIN\_ANG=180/QUAN\_ANG

ACC=FLTARR(kpts,BIN\_ANG,QUAN\_ANG) ;BIN\_ANG IS HOW MANY CELL FOR  
ANGLE

;SIZE OF ANGLE FOR ACCUMULATOR CELL

ANGLE=INDGEN(QUAN\_ANG) ;QUANTIZE

maxro=sqrt(n^2+m^2)

PRINT,'MAXRO',MAXRO

print,'angle',ANGLE

```

FOR I=0, KPTS-1 DO BEGIN
  FOR J=0, BIN_ANG-1 DO BEGIN

    ACC[i,J,*]=XPTS[i]*COS((ANGLE+J*QUAN_ANG-90)*!PI/180.)+YPTS[
i]*SIN((ANGLE+J*QUAN_ANG-90)*!PI/180.)

  ENDFOR
ENDFOR

HELP,ACC
MIN_RO=MIN(ACC)
MAX_RO=MAX(ACC)
RO_RANGE=ABS(MIN(ACC))+ABS(MAX(ACC))
BIN_RO=FIX(RO_RANGE/QUAN_RO)
RE=INTARR(BIN_RO)
;LOC_CO=INTARR(KPTS)
PRINT,'RO RANGE',RO_RANGE
print,'BIN_ANG=',BIN_ANG,' BIN_RO=',BIN_RO
;FOR I=0, BIN_RO-1 DO BEGIN
COUNTED=INTARR(BIN_ANG,BIN_RO)

  FOR J=0, BIN_ANG-1 DO BEGIN
    FOR K=0,BIN_RO-1 DO BEGIN
      RE=WHERE((ACC[* ,J,*] LT (QUAN_RO*(K+1)+MIN_RO)) AND (ACC[* ,J,*] GE
(QUAN_RO*(K)+MIN_RO)),COUNT)
      COUNTED[J,K]=COUNT
    ENDFOR
  ;print,'number',FIX((BIN_ANG)/10)
CASE J OF
0*FIX((BIN_ANG)/10) : BEGIN
WIDGET_CONTROL,info.destinationwin,GET_VALUE=tem
  WSET,tem & erase
  fname=FILEPATH('NUM9.TIF',SUBDIRECTORY='DATA')
  IM=READ_TIFF(FNAME)
  TV,IM
  END
1*FIX((BIN_ANG)/10) : BEGIN
WIDGET_CONTROL,info.destinationwin,GET_VALUE=tem
  WSET,tem & erase
  fname=FILEPATH('NUM8.TIF',SUBDIRECTORY='DATA')
  IM=READ_TIFF(FNAME)
  TV,IM
  END
2*FIX((BIN_ANG)/10) : BEGIN
WIDGET_CONTROL,info.destinationwin,GET_VALUE=tem
  WSET,tem & erase
  fname=FILEPATH('NUM7.TIF',SUBDIRECTORY='DATA')
  IM=READ_TIFF(FNAME)

```

```

    TV,IM
    END
3*FIX((BIN_ANG)/10) : BEGIN
WIDGET_CONTROL,info.destinationwin,GET_VALUE=tem
    WSET,tem & erase
    fname=FILEPATH('NUM6.TIF',SUBDIRECTORY='DATA')
    TV, READ_TIFF(FNAME)
    END
4*FIX((BIN_ANG)/10) : BEGIN
WIDGET_CONTROL,info.destinationwin,GET_VALUE=tem
    WSET,tem & erase
    fname=FILEPATH('NUM5.TIF',SUBDIRECTORY='DATA')
    TV, READ_TIFF(FNAME)
    END
5*FIX((BIN_ANG)/10) : BEGIN
WIDGET_CONTROL,info.destinationwin,GET_VALUE=tem
    WSET,tem & erase
    fname=FILEPATH('NUM4.TIF',SUBDIRECTORY='DATA')
    TV, READ_TIFF(FNAME)
    END
6*FIX((BIN_ANG)/10) : BEGIN
WIDGET_CONTROL,info.destinationwin,GET_VALUE=tem
    WSET,tem & erase
    fname=FILEPATH('NUM3.TIF',SUBDIRECTORY='DATA')
    TV, READ_TIFF(FNAME)
    END
7*FIX((BIN_ANG)/10) : BEGIN
WIDGET_CONTROL,info.destinationwin,GET_VALUE=tem
    WSET,tem & erase
    fname=FILEPATH('NUM2.TIF',SUBDIRECTORY='DATA')
    TV, READ_TIFF(FNAME)
    END
8*FIX((BIN_ANG)/10) : BEGIN
WIDGET_CONTROL,info.destinationwin,GET_VALUE=tem
    WSET,tem & erase
    fname=FILEPATH('NUM1.TIF',SUBDIRECTORY='DATA')
    TV, READ_TIFF(FNAME)
    END
9*FIX((BIN_ANG)/10) : BEGIN
WIDGET_CONTROL,info.destinationwin,GET_VALUE=tem
    WSET,tem & erase
    fname=FILEPATH('NUM0.TIF',SUBDIRECTORY='DATA')
    TV, READ_TIFF(FNAME)
    END
ELSE:
ENDCASE
ENDFOR
MAX_COUNT=MAX(COUNTED)

```

```

PRINT,'MAX COUNT =',MAX_COUNT
b = COUNTED(UNIQ(COUNTED, REVERSE(SORT(COUNTED))))

.....
,,,,,,,,,
THRESHOLD=B[13]
.....
,,,,,,,,,

ANG=(WHERE(COUNTED GE THRESHOLD,COUNT1) MOD (BIN_ANG))*quan_ang
RO=(WHERE(COUNTED GE THRESHOLD,COUNT2) / (BIN_ANG))*quan_ro
;LOC_CO=WHERE(COUNTED GE THRESHOLD)
print,'count1',count1,' count2',count2
ANG=ANG-90 & RO=RO+MIN_RO
REGION1=FLTARR(BIN_ANG,BIN_RO)
ANG_REGION=(WHERE(COUNTED GE THRESHOLD)MOD (BIN_ANG))
RO_REGION=(WHERE(COUNTED GE THRESHOLD)/ (BIN_ANG))

REGION1[ANG_REGION,RO_REGION]=255
REGION1[BIN_ANG-1,*]=0 & REGION1[BIN_ANG-2,*]=0 &
REGION1[BIN_ANG-3,*]=0 & REGION1[BIN_ANG-4,*]=0
  LABELED=LABEL_REGION(REGION1,/EIGHT)
  VALUES = LABELED(UNIQ(LABELED, SORT(LABELED)))
PRINT,'LABEL VALUES ',VALUES
NO_POINTS=SIZE(VALUES)
MED_ANG=FLTARR(NO_POINTS(1)-1) & MED_RO=FLTARR(NO_POINTS(1)-1)

FOR I=1, NO_POINTS(1)-1 DO BEGIN
LOC_ANG=WHERE(LABELED EQ VALUES[I]) MOD BIN_ANG
LOC_RO=WHERE(LABELED EQ VALUES[I]) / BIN_ANG
MED_ANG[I-1]=MEDIAN(LOC_ANG,/EVEN)*QUAN_ANG-90
MED_RO[I-1]=MEDIAN(LOC_RO,/EVEN)*QUAN_RO+MIN_RO
ENDFOR

LOADCT,0
;WINDOW,4
WIDGET_CONTROL,info.drawBE,GET_VALUE=tem
  WSET,tem & erase

PLOT,ANG_REGION*QUAN_ANG-90,RO_REGION*QUAN_RO+MIN_RO,PSYM=2,
XRANGE=[-90,90],$

  TITLE='FINDED POINTS IN PARAMETER COORDINATE',
Xtitle='Angle',Ytitle='Ro',xticklen=1,yticklen=1
  X=INDGEN(N)
  IMAGE=INTARR(N,M)
  WINDOW,6,TITLE='PLOT OF LINKED LINES',XPOS=600
  Y_TEM=FLTARR(N)

```

```

plot,x,Y_TEM,YRANGE=[0,m],background=255,color=0;,XRANGE=[-1
00,100],YRANGE=[-100,100]

PRINT,'ANGLE =',MED_ANG
PRINT,'RO  =',MED_RO
  FOR I=0,NO_POINTS(1)-2 DO BEGIN
    CASE SIN((MED_ANG[I])*!PI/180) OF
      0 : BEGIN Y=INDGEN(M)
          oplot,REPLICATE(MED_RO[I],N),Y,color=0
        END
      ELSE : BEGIN

Y=((MED_RO[I])-X*COS((MED_ANG[I])*!PI/180))/SIN((MED_ANG[I]) *!PI/180)
          oplot,X,y,color=0;,PSYM=3;,COLOR=I+1;IMAGE[* ,I]=Y
        END
      ENDCASE

    CASE COS((MED_ANG[I])*!PI/180) OF
      0 : BEGIN Y=REPLICATE(MED_RO[I],N)
          oplot,X,y,color=0;,PSYM=3;,COLOR=I+1;IMAGE[* ,I]=Y
        END
      ELSE : BEGIN
Y=((MED_RO[I])-X*COS((MED_ANG[I])*!PI/180))/SIN((MED_ANG[I]) *!PI/180)
          oplot,X,y,color=0;,PSYM=3;,COLOR=I+1;IMAGE[* ,I]=Y
        END
      ENDCASE

    ENDFOR
  PRINT,'SIZE OF SOURCE IMAGE',SIZE(SOURCE_IMAGE)
  HELP,IMAGE
  FOR I=0,NO_POINTS(1)-2 DO BEGIN
    CASE SIN((MED_ANG[I])*!PI/180) OF
      0 : BEGIN Y=INDGEN(N)
          IMAGE[REPLICATE(MED_RO[I],N),Y]=255
        END
      ELSE : BEGIN

Y=((MED_RO[I])-X*COS((MED_ANG[I])*!PI/180))/SIN((MED_ANG[I]) *!PI/180)
          IMAGE[X,Y]=255
        END
      ENDCASE

    CASE COS((MED_ANG[I])*!PI/180) OF
      0 : BEGIN Y=REPLICATE(MED_RO[I],N)
          IMAGE[X,Y]=255
        END
      ELSE : BEGIN
Y=((MED_RO[I])-X*COS((MED_ANG[I])*!PI/180))/SIN((MED_ANG[I]) *!PI/180)

```

```

        IMAGE[X,Y]=255
    END
ENDCASE
IF ABS(MED_ANG[I]) LT 10 AND MED_ANG[I] NE 0 THEN BEGIN
    Y=INDGEN(N)

    X=(MED_RO[I]-Y*SIN(MED_ANG[I]*PI/180))/COS(MED_ANG[I]*PI/180)
    IMAGE[X,Y]=255
    END
    IF ABS(MED_ANG[I]) GT 70 THEN BEGIN
        X=INDGEN(N)

        Y=((MED_RO[I])-X*COS((MED_ANG[I])*PI/180))/SIN((MED_ANG[I]) *PI/180)
        IMAGE[X,Y]=255
        END
    ENDFOR

```

```

    WIDGET_CONTROL,info.destinationwin,GET_Value=tem
    WIDGET_CONTROL,info.destinationwin,Set_UValue=IMAGE
    WSET, tem &erase
    TV,image
end

```

---