
Subject: Re: Best way to move a window

Posted by [Ned Horning](#) on Thu, 28 Feb 2002 15:47:02 GMT

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

Thanks Marc, this is pretty slick and it helps a lot. Unfortunately I just realized that I didn't post the second half of the window routine. After doing the "abs(tile_data - 180) lt 0.0001)" test I continue with:

```
s= 45
```

```
t=45
```

```
IF (((ABS(tile_data[x+1,y+1] - 45) le s) AND (ABS(tile_data[x-1,y-1] -  
tile_data(x+1,y+1) -180) le t)) OR $
```

```
((ABS(tile_data[x+1,y] - 90) le s) AND (ABS(tile_data[x-1,y] -  
tile_data(x+1,y) -180) le t)) OR $
```

```
((ABS(tile_data[x+1,y+1] - 135) le s) AND (ABS(tile_data[x-1,y+1] -  
tile_data(x+1,y-1) -180) le t)) OR $
```

```
((ABS(tile_data[x,y-1] - 180) le s) AND (ABS(ABS(tile_data[x,y+1] -  
tile_data(x,y-1)) -180) le t))) THEN $
```

```
out_tile[y*num_tile_samples+x] = 1 ELSE $
```

```
out_tile[y*num_tile_samples+x] = 0
```

What this does is compares opposing pixels (top and bottom, left and right, upper left lowerright, upper right and lower left) and I'm trying to see how to use the where/convolution logic for this and not having much luck.

Here is the entire code block :

```
FOR l=0, num_tile_samples-1 DO BEGIN
```

```
  out_tile[l] = 0
```

```
ENDFOR
```

```
FOR y=1, num_tile_lines - 2 DO BEGIN
```

```
  out_tile[y*num_tile_samples] = 0
```

```
  FOR x=1, num_tile_samples - 2 DO BEGIN
```

```
    IF (((tile_data[x,y] gt 179.99999) AND (tile_data[x,y] lt  
180.0001)) OR $
```

```
      ((tile_data[x+1,y] gt 179.99999) AND (tile_data[x+1,y] lt  
180.0001)) OR $
```

```
      ((tile_data[x-1,y] gt 179.99999) AND (tile_data[x-1,y] lt  
180.0001)) OR $
```

```
      ((tile_data[x,y+1] gt 179.99999) AND (tile_data[x,y+1] lt  
180.0001)) OR $
```

```
      ((tile_data[x,y-1] gt 179.99999) AND (tile_data[x,y-1] lt  
180.0001)) OR $
```

```
      ((tile_data[x+1,y+1] gt 179.99999) AND (tile_data[x+1,y+1] lt  
180.0001)) OR $
```

```

        ((tile_data[x-1,y-1] gt 179.99999) AND (tile_data[x-1,y-1] lt
180.0001)) OR $
        ((tile_data[x-1,y+1] gt 179.99999) AND (tile_data[x-1,y+1] lt
180.0001)) OR $
        ((tile_data[x+1,y-1] gt 179.99999) AND (tile_data[x+1,y-1] lt
180.0001))) THEN $
        out_tile[y*num_tile_samples+x] = 0 ELSE BEGIN

        IF (((ABS(tile_data[x+1,y+1] - 45) le s) AND
(ABS(tile_data[x-1,y-1] - tile_data(x+1,y+1) -180) le t)) OR $
        ((ABS(tile_data[x+1,y] - 90) le s) AND
(ABS(tile_data[x-1,y] - tile_data(x+1,y) -180) le t)) OR $
        ((ABS(tile_data[x+1,y+1] - 135) le s) AND
(ABS(tile_data[x-1,y+1] - tile_data(x+1,y-1) -180) le t)) OR $
        ((ABS(tile_data[x,y-1] - 180) le s) AND
(ABS(ABS(tile_data[x,y+1] - tile_data(x,y-1)) -180) le t))) THEN $
        out_tile[y*num_tile_samples+x] = 1 ELSE $
        out_tile[y*num_tile_samples+x] = 0
        ENDELSE
    ENDFOR
    out_tile[y*num_tile_samples+x] = 0
ENDFOR
FOR l=0, num_tile_samples-1 DO BEGIN
    out_tile[y*num_tile_samples+l] = 0
ENDFOR

```

```

>
> Indeed there is:
>
> ;;here we go -----
> kernel=bytarr(3,3)
> kernel[*]=1
>
> boolArr=bytarr(num_tile_samples,num_tile_samples)
> w=where(abs(tile_data - 180.0) lt 0.0001)
> ;; check (if necessary)
> if w[0] eq -1 then return ;; or whatever
> boolArr[w]=1
>
> res=convol(boolArr,kernel,/EDGE_TRUNCATE)
>
> ;; these are the indices you want
> w1=where(res ge 1)
>
> ;; w9=where(res ge 9) ;; these are the ones if you would AND instead of
> OR
> ;; in the example above

```

>
> out_tile[w1]=0
> ;;finish -----
>
> you may want to check out also the dilate or erode function...
>
> cheers,
> marc
