Subject: Removing stray clusters of points
Posted by Todd Clements on Fri, 01 Jun 2001 21:52:03 GMT
View Forum Message <> Reply to Message

Hi all again...

Thanks again Craig for your response on the ellipse fitting, it does most of what I need. However, because our images are so broad, the edge detection method you suggested doesn't work. The roberts() and sobel() routines DO enhance the edge, but not quite enough, and they leave some stray pixel clusters.

So, what I did instead was write an algorithm that simply finds the literal edge of the image (after edge enhancement) coming from the left and right. This gives me a rough outline of a circle that I can use the ellipse fitting program. This works well except sometimes there are a few sets of stray pixels clusters that interfere with finding the edge of the image. I'd love to be able to remove these stray clusters.

I found the sigma_filter function in one of the on-line libraries, but it doesn't seem to quite work no matter what box size or sigma I use. So, right now what I've done is the following. I look for small boxes inside large boxes where there are no additional pixels around the small box. This does what I want, but is slow. As you can see from the code, these are really isolated patches, which should be easier to get rid of!

```
-----
;; znew is the image data
s = size( znew, /dimensions )
newSum = fltarr( s[0], s[1] )
newSum1 = fltarr( s[0], s[1] )
bs = 8
ss= 4
for i=-ss, ss do for j=-ss,ss do newSum1 = newSum1 + shift( znew, i, j )
for i=-bs, bs do for j=-bs,bs do newSum = newSum + shift( znew, i, j )
junk = where( newsum1 eq newSum )
if( junk[0] ne -1 ) then $
  znew[junk] = 0
-----
```

znew is usually 512x512, but this is still a fairly slow process, and I'm sure there's a better way of doing it. I don't have to worry about edges of the array, since there is generally no data near our array edges.

Any thoughts regarding the problem (or just general life philosophy =) are appreciated.

TIA,
Todd