Subject: Index of a sector
Posted by Pavel A. Romashkin on Wed, 04 Apr 2001 23:18:11 GMT
View Forum Message <> Reply to Message

Hi,

I tried to solve this yesterday, but it gave me a headache. I tried again today, but the headache is back and I did not get any closer to the solution. It is very simple. Well after all, what can you ask from a Mac user.

If you have a (square) 2D array, let's say, DIST(200, 200), how to obtain the index of points enclosed by a sector of a given radius, drawn from the corner of the array (point [0, 0])? Foe instance, radius 50?
So, I will have

0,1,2,...50
101,102,...,149 (or so)
201,202,...
...
501,...?

I thought to try to cut all corners and use IDLanROI, but for some reason, I think that JD or Craig have the answer ready. Besides, David tried anROI... how can I dream of succeeding where the Titan (read: Coyote) failed?
Thanks,
Pavel

Subject: Re: Index of a sector
Posted by Craig Markwardt on Thu, 05 Apr 2001 18:42:34 GMT
View Forum Message <> Reply to Message

"Pavel A. Romashkin" <pavel.romashkin@noaa.gov> writes:

> Hi,
>
> I tried to solve this yesterday, but it gave me a headache. I tried
> again today, but the headache is back and I did not get any closer to
> the solution. It is very simple. Well after all, what can you ask from a
> Mac user.
>
> If you have a (square) 2D array, let's say, DIST(200, 200), how to
> obtain the index of points enclosed by a sector of a given radius, drawn
> from the corner of the array (point [0, 0])? Foe instance, radius 50?
> So, I will have
>
> 0,1,2,...50

> 101,102,...,149 (or so)
> 201,202,...
> ...
> 501,...?

Complex geometric selections can be easy if you do the right thing. Pavel, I'm sure that Med's suggestion can work for you but I thought I would expand on it a little more for the general case.

You start by labelling your pixels in X and Y. You might have physical labels attached to each pixel, or they could just be pixel numbers.

```
  x = findgen(200) & y = findgen(200)  ;; Just an example here!
```

Now expand these vectors into matrices along their respective axes:

```
  xx = x # (fltarr(200)+1)   ;; Now XX labels every pixel in image
  yy = (fltarr(200)+1) # y   ;; as does YY
```

Now compute anything you want in this coordinate system, say radius:

```
  rr = sqrt(xx^2 + yy^2)     ;; could compute theta as, th = atan(yy, xx)
```

And then do your selection:

```
  wh = where(rr LE 50 AND xx GE 0 AND yy GE 0)
```

You are done. As you can see you can form very complex expressions involving the radius and angle, and if you have multiple centers this can be accomodated too. The above expansions of X into XX and Y into YY are also the basis of fitting images in MPFIT2DFUN.

Good luck,
Craig

--
 -- --------------
Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 -- --------------

Subject: Was: Index... Now: Vectorize, huh?
Posted by Pavel A. Romashkin on Thu, 05 Apr 2001 19:24:25 GMT
View Forum Message <> Reply to Message

Craig and Med,

I appreciate it! This is exactly what I needed. My problem is the lack
of matrix operations knowledge. Craig's generic solution is exactly what
I expected to see from Craig :-)

By the way. We all are big on vectorizing things in IDL. But look at this:

```
IDL> a = test(2000)
      1.4968959
IDL> a = test(2000, /v)
      3.2976190
```

where TEST is below. I don't even mention that /VEC causes extremely
high memory usage and gets totally out of hand on my system if S > 5000
or so.

```
;*****************************
pro test, s, vec=vec
start = systime(1)
x = findgen(s)
a = fltarr(s, s)
if keyword_set(vec) then begin
a = sqrt(transpose(rebin(x, s, s))^2 + rebin(x, s, s)^2)
endif else begin
for i = 0, s-1 do begin
a[0, i] = sqrt(x^2 + i^2.)
endfor
endelse
print, systime(1) - start
;return, a
end
;*****************************
```