Subject: Re: Help requested in eradicating FOR loops
Posted by Vince Hradil on Fri, 14 Dec 2007 16:24:29 GMT
View Forum Message <> Reply to Message

On Dec 14, 10:06 am, dplat...@gmail.com wrote:
> Dear IDL users,
>
> I would really appreciate some help in removing a nested FOR loop.
>
> FOR x=0, image_width-1 DO BEGIN
> FOR y=0, image_height-1 DO BEGIN
> ; Write the current pixel value
> results[(x*mtf_region_height)+y, 1] = image[x,y]
> ; Calculate the distance of the current pixel from the
> line defined
> by the
> ; two points (x1, y1) and (x2, y2).
> results[(x*image_height)+y, 0] = ((y1-y2)*x + (x2-
> x1)*y + (x1*y2 -
> x2*y1)) / SQRT((x2-x1)^2 + (y2-y1)^2)
> ENDFOR
> ENDFOR
>
> 'image' is a 2d image array
> I want to work out the distance of each pixel from a line that is
> defined by the two points (x1, y1), (x2, y2) and store this distance,
> together with the pixel value in a 'results' array. The 'results'
> array has the dimensions [(image_height*image_width), 2].
>
> The above bit of code works but is a bit slow.
>
> Any advice would be much appreciated.
>
> Thanks,
>
> David
> Northampton, UK

Off the top of my head:
1-reform the points to match the size of the image: x1 =
reform(x1,image_height,image_width)
2-make in "index" array: idx = findgen(image_height,image_width)
3-make x-index and y-index arrays: x = idx mod image_height & y = idx/
image_height (or maybe those should be image_width?)
4-do the calculation: distance = ((y1-y2)*x + (x2-x1)*y + (x1*y2-
x2*y1)) / SQRT((x2-x1)^2 + (y2-y1)^2)
5-put together the results matrix: results =
[[reform(image,image_height*image_width)],[distance]] (I'd have to

check the brackets, I always do)

Subject: Re: Help requested in eradicating FOR loops
Posted by dplatten on Tue, 18 Dec 2007 11:00:45 GMT
View Forum Message <> Reply to Message

> Off the top of my head:
> 1-reform the points to match the size of the image: x1 =
> reform(x1,image_height,image_width)
> 2-make in "index" array: idx = findgen(image_height,image_width)
> 3-make x-index and y-index arrays: x = idx mod image_height & y = idx/
> image_height (or maybe those should be image_width?)
> 4-do the calculation: distance = ((y1-y2)*x + (x2-x1)*y + (x1*y2-
> x2*y1)) / SQRT((x2-x1)^2 + (y2-y1)^2)
> 5-put together the results matrix: results =
> [[reform(image,image_height*image_width)],[distance]] (I'd have to
> check the brackets, I always do)

Thanks for the reply - as I understand it I need to make an array to
hold x pixel positions and one to hold y pixel positions. For a 4 x 6
array they would look like this:

x locations: [0,1,2,3, 0,1,2,3, 0,1,2,3, 0,1,2,3, 0,1,2,3, 0,1,2,3]
y locations: [0,0,0,0, 1,1,1,1, 2,2,2,2, 3,3,3,3, 4,4,4,4, 5,5,5,5]

I can make the x array by doing this:

test_image = findgen(4, 6) ; a dummy test "image"
temp = findgen(4 * 6)
x_locations = temp mod 4

but I am having problems creating the y locations array. Some help
would be appreciated.

Thanks, David