
Subject: Re: Line index

Posted by [Robert Moss](#) on Mon, 25 Mar 1996 08:00:00 GMT

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

I may have missed your initial question, but if your looking for the array indices along a line in an image, the following snippet of code seems to work fine. The two endpoints are (x,y) and (x1,y1), and the image is in the variable pimage.

```
dx = float(x1-x) ;delta x
dy = float(y1-y) ;delta y
n = sqrt( dx^2 + dy^2 ) ;length
sy =(y1-y)/n
sx =(x1-x)/n
;
xx = lindgen(n+1)*sx+x ;X indices, make into longwords.
yy = lindgen(n+1)*sy+y ;Y indices
sz = size( pimage ) ; image size
if !order ne 0 then yy=sz(2)-1-yy ;reverse y indices?
ans=pimage(xx,yy) ; image data along the line
```

You may want to go some fiddling with the (n+1) bit depending on how you want to handle fractional pixels.

By the way, this is what I use when I want a "profile" of an image, i.e I want the image data values along the line connecting two points. The length reported here is equal to the length of the line (within pixel resolution, your milage may vary).

Note that this is different than the IDL supplied library funtction called "profile" which actually returns a list of values that are the length of the x or y projection of the line connecting two points, depending on which is longer. I did not find the builtin "profile" results acceptable for my purposes.

Robet M. Moss, Ph.D.
Texaco Inc.
mossrm@texaco.com

This may not reflect the views of Texaco Inc.

David Foster wrote:
>

```
> pit@asterix.kis.uni-freiburg.de (Peter Suetterlin) wrote:
>>
>> In article <31499FB9.2F6D@dlr.de>,
>>   Hermann Mannstein <H.Mannstein@dlr.de> writes:
>>
>>> is there a robust function which returns the indices within an image
>>> (like the where function) belonging to a line defined by two or more
>>> points.
>>
>> for shure there is, even built-in IDL:
>> Index=polyfillv(xx,yy,xsize,ysize)
>> Read the manpage for further reference.
>>
>
> According to the "manpage", POLLYFILLV() expects at least 3
> points so it would never work for a simple line. Also, it
> connects each successive line and the first with the last,
> to make a polygon, and then returns the indices of the
> points within this polygon.
>
> You wouldn't be getting the indices of the points on the line(s),
> but of the points within the polygon defined by the points.
>
> Dave Foster
> foster@bial1.ucsd.edu
```
