Subject: Re: Center of mess
Posted by Craig Markwardt on Tue, 05 Aug 2003 05:23:17 GMT
View Forum Message <> Reply to Message

Thomas Launey <t_launey@brain.riken.go.jp> writes:

> Hello,
> I am trying to calculate the centroid of a small object in an image.
> The procedure that I wrote is very standard; I found the same one in
> David Fannning's archive of this NG and in IDLastro. However, the result
> that I get does not seem right. In the code below, the central pixel of a
> 5x5 image is the peak so I would expect the centroid to be at (2.5,2.5)
> but the result is [2.0,2.0]. I agree that pixel "2" is the correct answer
> but I would expect that the correct coordinates should be the center of
> pixel "2" (I.e.: [2.5,2.5]). I am using this centroid function to refine
> peak coordinates in a 2D cross-correlation so I need this 0.5 pixel
> precision.
...
> ;*** collapse the array on Y axis
> Y=Total(total(array,1,/double)*(dindgen(sizarr[1])))/totalar r
> ;*** collapse the array on X axis
> X=Total(total(array,2,/double)*(dindgen(sizarr[0])))/totalar r

Hmm, the problem is that you are assigning pixel values with DINDGEN,
which by default will assign 0.0 to pixel 0, 1.0 to pixel 1, and so
on. You should add 0.5 to your pixel values, if you intend that the
left edge of the pixel is 0.0, the right edge is 1.0, and the center
is 0.5, etc.

```
Y=Total(total(array,1,/double)*(dindgen(sizarr[1]) + 0.5))/totalarr
```

and so on.

Good luck,

Craig

Subject: Re: Center of mess
Posted by Thomas Launey on Tue, 05 Aug 2003 09:43:05 GMT
View Forum Message <> Reply to Message

In article <onwudshe4a.fsf@cow.physics.wisc.edu>,
craigmnet@cow.physics.wisc.edu says...

>
> Hmm, the problem is that you are assigning pixel values with DINDGEN,

> which by default will assign 0.0 to pixel 0, 1.0 to pixel 1, and so
> on. You should add 0.5 to your pixel values, if you intend that the
> left edge of the pixel is 0.0, the right edge is 1.0, and the center
> is 0.5, etc.
>
> Y=Total(total(array,1,/double)*(dindgen(sizarr[1]) + 0.5))/totalarr
>
> and so on.
>
> Good luck,
>
> Craig
>
Actually, my solution was to add 0.5D to the returned X and Y. Still, I
am concerned that this procedure (without the "+0.5") seems to be quite
widespread in the IDL community to find centroid. Am I misunderstanding
the meaning of the returned values or is it a bug in the procedure ?
Thanks,
Thomas
--
Thomas LAUNEY
Lab. Memory and Learning, RIKEN BSI

Subject: Re: Center of mess
Posted by James Kuyper on Tue, 05 Aug 2003 13:45:30 GMT
View Forum Message <> Reply to Message

Thomas Launey wrote:
>
> In article <onwudshe4a.fsf@cow.physics.wisc.edu>,
> craigmnet@cow.physics.wisc.edu says...
>
>>
>> Hmm, the problem is that you are assigning pixel values with DINDGEN,
>> which by default will assign 0.0 to pixel 0, 1.0 to pixel 1, and so
>> on. You should add 0.5 to your pixel values, if you intend that the
>> left edge of the pixel is 0.0, the right edge is 1.0, and the center
>> is 0.5, etc.
>>
>> Y=Total(total(array,1,/double)*(dindgen(sizarr[1]) + 0.5))/totalarr
>>
>> and so on.
>>
>> Good luck,
>>
>> Craig
>>

> Actually, my solution was to add 0.5D to the returned X and Y. Still, I
> am concerned that this procedure (without the "+0.5") seems to be quite
> widespread in the IDL community to find centroid. Am I misunderstanding
> the meaning of the returned values ...

Yes. Craig explained the correct interpretation of the returned values.
It is, in a certain sense, the natural interpretation, because it's the
one that is produced by one of the simplest version of the algorithm.
One alternative that would appeal to someone from a Fortran background
would be to assign 1.0 to the center of the first row/column, rather
than 0.0.
