

---

Subject: Re: Derivative

Posted by [Vince Hradil](#) on Tue, 01 Jul 2008 19:36:13 GMT

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

---

On Jul 1, 2:11 pm, w...@bao.ac.cn wrote:

> On Jul 2, 3:05 am, Vince Hradil <hrad...@yahoo.com> wrote:

>

>

>

>> On Jul 1, 1:58 pm, w...@bao.ac.cn wrote:

>

>>> I have a 2-D array (a image) A. I want to know its partial

>>> derivatives along X direction and Y direction.

>>>  $dA(x,y)/dx=B$

>>>  $dA(x,y)/dy=C$

>>> B and C should also be 2-D array.

>>> I want to know if IDL has some function or operator can do this job

>>> directly.

>>> If it has not those I wanted, then I must write some program according

>>> to "Spline and Lagrange".

>>> I am not a foolish boy. But I am a lazy boy.

>>> Who can tell me if IDL can do this job for me?

>>> Thanks!

>

>> have you tried `deriv()`? Have you tried:

>> IDL> ?

>

> yes,I have used `deriv()` before.But it only works for one dimension.

> If I use it to differentiate a 2-D image along one direction, then I

> must use "for i= , do begin .....endfor". That is too slow.

> Could you give me more hints?

Ahh... sorry - I was in a bad mood.

This is what I've done (as part of a Canny routine, btw):

```
isize = size(image,/dimensions)
```

```
ncol = isize[0]
```

```
nrow = isize[1]
```

```
grad = fltarr(ncol,nrow,4)
```

```
filter = double([ [-1,-2,-3,-2,-1], [0,0,0,0,0], [1,2,3,2,1] ])
```

```
grad[* , 0] = convol(image,transpose(filter),/center,/edge_truncate) ;
```

```
Horizontal
```

```
grad[* , 1] = convol(image,filter,/center,/edge_truncate) ; Vertical
```

```
grad[* , 2] = sqrt( grad[* , 0]^2 + grad[* , 1]^2 ) ; Magnitude
```

```
grad[* , 3] = atan( grad[* , 1], grad[* , 0] ) ; Direction
```

I hope this helps you get started. I might have `horiz/vert`

reversed... it's been a while since I actually used this 8^)

---