Subject: Re: Gradient of an Image in IDL
Posted by David Foster on Tue, 03 Jun 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Brad Cadle wrote:
>
> HI all,
> I was wondering if someone knows of an optimized IDL routine to
> compute the gradient of an image, I ported one over from C where the
> gradient is generated in fourier space, but I was interested to find
> if one was optimized for IDL.

One alternative is to use CALL_EXTERNAL() to interface to your (already optimized) C routine, which would require slight modifications to the routine. I can mail you examples of using this method if you're interested.

Dave
--

~~~~~~ ~~~~~~
David S. Foster Univ. of California, San Diego
Programmer/Analyst Brain Image Analysis Laboratory
foster@bial1.ucsd.edu Department of Psychiatry
(619) 622-5892 8950 Via La Jolla Drive, Suite 2200
La Jolla, CA 92037
~~~~~~ ~~~~~~

"I have this theory that if we're told we're bad,
then that's the only idea we'll ever have.
But maybe if we are surrounded in beauty,
someday we will become what we see." - Jewel Kilcher

Subject: Re: Gradient of an Image in IDL
Posted by Hermann Mannstein on Thu, 05 Jun 1997 07:00:00 GMT
View Forum Message <> Reply to Message

David Foster wrote:
>
> Brad Cadle wrote:
>>
>> HI all,
>> I was wondering if someone knows of an optimized IDL routine to
>> compute the gradient of an image, I ported one over from C where the
>> gradient is generated in fourier space, but I was interested to find
>> if one was optimized for IDL.
>

> One alternative is to use CALL_EXTERNAL() to interface to your (already
> optimized) C routine, which would require slight modifications to the
> routine. I can mail you examples of using this method if you're
> interested.
>
> Dave

I'm not shure that it is really optimized, and there are different ways
to
define the gradient, but the following works on 2-d images and gives a
centered
absolute value and the angle in degree - i.e. it uses all surrounding
pixels.
The convolution with an 3X3 array is quite fast.

```
function sin_kern0
a=[ [
 0.7071067812,1.,0.7071067812],[0.,0.,0.],[-0.7071067812,-1., -0.7071067812]
]
return,a
end

function cos_kern0
a = [
 [-0.7071067812,0.,0.7071067812],[-1.,0.,1.],[-0.7071067812,0 .,0.7071067812]
]
return,a
end

pro gradient,input,gradient,ang
r2deg = -90./!pi

cu = float(input)
a = sin_kern0()
s = convol(cu,a)
a = cos_kern0()
c = convol(cu,a)
gradient= sqrt(s*s + c*c)
ang    = atan(s,c) * r2deg
return
end
```

--
Regards,

~~~~~~
~~~~~~~~~~~~~~~~~~~~

Hermann Mannstein Tel.: +49 8153 28-2503
Institut fuer Physik der Atmosphaere or -2558
DLR - Oberpfaffenhofen Fax.: +49 8153 28-1841
Postfach 1116 \ mailto:H.Mannstein@dlr.de
D-82230 Wessling \ 0 http://www.op.dlr.de/~pa64
Germany ________\/|________
~~~~~~`---------\---------'~ ~~~~~~~~~~~~~~~~~~~~
\

~~~~~~