Subject: Re: Gradient of an Image
Posted by meron on Fri, 24 Jan 2003 07:05:43 GMT
View Forum Message <> Reply to Message

In article <3E30D896.7000602@ee.uwa.edu.au>, Thomas Gutzler <tgutzler@ee.uwa.edu.au> writes:
> Hi,
>
> is it true that IDL currently has no function to calculate the gradient
> of an image ?
> Google found this for me:
> http://groups.google.com/groups?q=gradient+image+group:comp.lang.idl-pvwave&hl=en&lr=&ie=UTF-8&oe=UTF-8&selm=3396D128.4CE0%40dlr.de&rnum=1
> and I am about to test, if this is doing the same as "Digital Image
> Processing" by Gonzales/Woods sais on page 418ff.
> Would be interesting to know, if anybody else has diffent (quicker?)
> solutions.
>
If it is just the absolute value of the gradient you're after, then
I've a function like this (written many years ago). And it can easily
be modified to split the components of the gradient. The function is
called ABGRAD and you'll find it in the IDL users contributions page,
in my library (MIDL).

Mati Meron | "When you argue with a fool,
meron@cars.uchicago.edu | chances are he is doing just the same"

Subject: Re: Gradient of an Image
Posted by Thomas Gutzler on Fri, 24 Jan 2003 08:23:45 GMT
View Forum Message <> Reply to Message

Hi Mati,

meron@cars3.uchicago.edu wrote:
> In article <3E30D896.7000602@ee.uwa.edu.au>, Thomas Gutzler <tgutzler@ee.uwa.edu.au> writes:
>
>> Hi,
>>
>> is it true that IDL currently has no function to calculate the gradient
>> of an image ?
>> Google found this for me:
>> http://groups.google.com/groups?q=gradient+image+group:comp.lang.idl-pvwave&hl=en&lr=&ie=UTF-8&oe=UTF-8&selm=3396D128.4CE0%40dlr.de&rnum=1
>> and I am about to test, if this is doing the same as "Digital Image

>> Processing" by Gonzales/Woods sais on page 418ff.
>> Would be interesting to know, if anybody else has diffent (quicker?)
>> solutions.
>>
>
> If it is just the absolute value of the gradient you're after, then
> I've a function like this (written many years ago). And it can easily
> be modified to split the components of the gradient. The function is
> called ABGRAD and you'll find it in the IDL users contributions page,
> in my library (MIDL).

I figured out, that Hermann Mannsteins function does what I want. It calculates a gradient of an image using the 'Sobel operator' and it does it _very_ much faster than my testfunction did (It simply went through the array in 2 for-loops and multiplied the subarray with the kernel, summed the results and stored them in the final gradient-array).
convol rox :>
Just needs a Boundary-expansion to get better values at the bounds.

If anybody wants to have the code, just drop an email.

thanks anyway,
Tom

Subject: Re: Gradient of an Image
Posted by meron on Fri, 24 Jan 2003 08:45:14 GMT
View Forum Message <> Reply to Message

In article <3E30F811.8020209@ee.uwa.edu.au>, Thomas Gutzler <tgutzler@ee.uwa.edu.au> writes:
> Hi Mati,
>
> meron@cars3.uchicago.edu wrote:
>> In article <3E30D896.7000602@ee.uwa.edu.au>, Thomas Gutzler <tgutzler@ee.uwa.edu.au> writes:
>>
>>> Hi,
>>>
>>> is it true that IDL currently has no function to calculate the gradient
>>> of an image ?
>>> Google found this for me:
>>> http://groups.google.com/groups?q=gradient+image+group:comp.lang.idl-pvwave&hl=en&lr=&ie=UTF-8&oe=UTF-8&selm=3396D128.4CE0%40dlr.de&rnum=1
>>> and I am about to test, if this is doing the same as "Digital Image
>>> Processing" by Gonzales/Woods sais on page 418ff.
>>> Would be interesting to know, if anybody else has diffent (quicker?)

>>> solutions.
>>>
>>
>> If it is just the absolute value of the gradient you're after, then
>> I've a function like this (written many years ago). And it can easily
>> be modified to split the components of the gradient. The function is
>> called ABGRAD and you'll find it in the IDL users contributions page,
>> in my library (MIDL).
>
> I figured out, that Hermann Mannsteins function does what I want. It
> calculates a gradient of an image using the 'Sobel operator' and it does
> it _very_ much faster than my testfunction did (It simply went through
> the array in 2 for-loops and multiplied the subarray with the kernel,
> summed the results and stored them in the final gradient-array).
> convol rox :>

Sure.

> Just needs a Boundary-expansion to get better values at the bounds.
>
Unless you've information regarding what the boundary should be, it is
best to either force it to zero or to establish values by continuity.

Mati Meron | "When you argue with a fool,
meron@cars.uchicago.edu | chances are he is doing just the same"

Subject: Re: Gradient of an Image
Posted by David Fanning on Fri, 24 Jan 2003 14:30:30 GMT
View Forum Message <> Reply to Message

Thomas Gutzler (tgutzler@ee.uwa.edu.au) writes:
>
> I figured out, that Hermann Mannsteins function does what I want. It
> calculates a gradient of an image using the 'Sobel operator' and it does
> it _very_ much faster than my testfunction did (It simply went through
> the array in 2 for-loops and multiplied the subarray with the kernel,
> summed the results and stored them in the final gradient-array).
> convol rox :>

Yes, it looks to me to do *exactly* what the SOBEL function
does. The other gradient operator found in IDL is the ROBERTS function.

Cheers,

David

--

David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155