Subject: Gradient Function
Posted by jyoung on Mon, 20 May 1996 07:00:00 GMT
View Forum Message <> Reply to Message

I am new to IDL and I am looking for a function that might speed
up a procedure I am working on. I have a two dimensional array that has
some data points mostly low values, but in places it jumps rather quickly
to high values. I want to put zeros at the constant places and ones at
the quick changes.
It was suggested to me that I should try some kind of gradient
function. I can probably create my own for IDL, but I wanted to know if
there was any kind of function that might help in my efforts. I think the
DERIV function might help, but maybe there is a more powerful function to
examine these changes within the array.
Lastly what I have created causes a shift of the data by 1 element
in the new array...any suggestions on how to fix this?

Jason Young
EPSCoR Research Assistant
W.V.I.T.
Montgomery, WV 25136

phone: 304-442-1021

e-mail: jyoung@olie.wvitcoe.wvnet.edu

Subject: Re: Gradient Function
Posted by ralfu on Tue, 21 May 1996 07:00:00 GMT
View Forum Message <> Reply to Message

In article <19960520.114958.988283.NETNEWS@WVNVM.WVNET.EDU>,
jyoung@olie.wvitcoe.wvnet.edu (Jason Young) writes:
> I am new to IDL and I am looking for a function that might speed
> up a procedure I am working on. I have a two dimensional array that has
> some data points mostly low values, but in places it jumps rather quickly
> to high values. I want to put zeros at the constant places and ones at
> the quick changes.
> It was suggested to me that I should try some kind of gradient
> function. I can probably create my own for IDL, but I wanted to know if
> there was any kind of function that might help in my efforts. I think the
> DERIV function might help, but maybe there is a more powerful function to
> examine these changes within the array.

I looked very low values in a 2D array and did this in the following
way: first smoothing the array with SMOOTH, substracting this from
the real data, then looking in the remaining array for sharp features
with the edge detecting SOBEL filter. Then I go with a loop through

the resulting index array and locate the rectangular subarrays,
which contain exactly one of the sharp minima I look for, this I
get with MIN and WHERE.
Perhaps not the most elegant method, but it works for me.

Regards, Ralf

--
Ralf Utermann
_____ _____
Universitaet Augsburg (Germany), Institut fuer Physik
Memmingerstr.6 "Speaking for me and nobody else"
D-86135 Augsburg Phone: +49-821-598-3231
SMTP: Ralf.Utermann@Physik.Uni-Augsburg.DE Fax: -3222