Subject: Re: IDL's built-in function DILATE and ERODE doesn't work as described in help

Posted by gqshen2008@gmail.com on Mon, 16 Oct 2006 20:29:46 GMT

View Forum Message <> Reply to Message

I know this topic has gone beyond my initial question, but I have got
a response from ITTVIS support.
In IDL, if you don't set the optional argument(s), Nx, Ny or Nz, IDL
thinks that Nx = DIMENSION(X) /2, which means that
DILATE([0, 1, 1, 0], [1, 1])
has the same effect as
DILATE([0, 1, 1, 0], [1, 1], 1)
which means that for every 1 in the data, it will align with k[1], so
the output is [1, 1, 1, 0].
If you want to get the same result as in their help, you should
explicitly set the Nx to 0.
One interesting thing is that you can set Nx to 2, 3, 5 or even -1,
-2, etc.

Another interesting finding is that if you have:
DILATE([0, 1, 1, 0], [1, 1], 3)
My initial guess is that it will output [1, 0, 0, 0], however, IDL
will not replace 1 with [1, 1, 0, 0] unless any kernel's pixle overlays
with the original data. So the real result is [0, 0, 0, 0].

Notice the sudden jump of output from [0, 0, 0, 0]:
DILATE([0, 0, 1, 1], [1, 1], 3)
will output [1, 1, 0, 0].

Gongqin Shen wrote:

> For example, if you have the data as a = [0, 1, 1, 0] and kernel as k
> = [1, 1], according to the help provided by IDL, the result of running
> the code:
> result = DILATE(a, k)
> will be [0, 1, 1, 0], however, IDL's output is [1, 1, 1, 0].
> ERODE performs in a similar way. Does that mean the help is actually
> broken?