Subject: Re: IDL's built-in function DILATE and ERODE doesn't work as described in help
Posted by Karsten Rodenacker on Thu, 12 Oct 2006 17:06:37 GMT
View Forum Message <> Reply to Message

In fact I sent mail only to Karl. Possibly the things are of interest for others, although there were not so many math. morph adepts:

Mathematical Morphology (MM) in IDL

I consider the actual state of the routines derived from math. morph. in IDL as more or less useless. MM is in fact a methodology which could be VERY helpful for image processing, provided the implementation does not it break out, what is the case in IDL. My remark comprise the actual routines
 ERODE
 DILATE
 MORPH_DISTANCE
 THIN
 LABEL_REGION
 WATERSHED
as well as the derivatives
 MORPH_CLOSE
 MORPH_GRADIENT
 MORPH_HITORMISS
 MORPH_OPEN
 MORPH_THIN
 MORPH_TOPHAT
which suffer all from the unsufficient implementation.

Main point which should be improved is the processing of the array border points. I had some time ago already exchange about that but seemingly the rsi counterpart did nothing know about MM. However, I am interested in MM so I try it again!

ERODE/DILATE:
Basically a result of ERODE/DILATE is the logical AND/OR rspw. min/max of all pixels of the structuring element (st.e.). A border point is a point where the st.e. does contain points outside the data array. What should be done with these points? The worst is to say don't use these points, embed the data in a sufficient large data array. This is the way IDL uses. This approach is worse since the successive application of these operators is not possible or to slow. Additionally erosion and dilation are dual, with other words both should behave in the same way:
 ERODE(x,st.e.) = NOT DILATE(NOT x, st.e)
What to do: There should be a switch, a system variable !MM.edge saying:
 outside the data matrix everything is 1b/true (binary) or max(data) (gray) OR
 outside the data matrix everything is 0b/false (binary) or min(data)

(gray)
Now all border points have well defined values and duality of erosion and dilation can be preserved, no embedding is necessary, successive operations are possible and the whole MM apparatus can be implemented. See e.g. the book "Morphological Image Analysis", P. Soille, Springer

MORPH_DISTANCE
This routines suffers from the same problem. If an object is touching the border in the actual implementation border points are considered as object border points. Using the above mentioned !MM.edge morph_distance could deliver border independent results. and could be applied to the complement of the image without poblem, which is often necessary!
This routine is even usable for quick erosions and dilations by thresholding the distance map as I did earlier. The neighbor_sampling keyword selects the corresponding st.e..

THIN
is quick but in MM terms completely useless, since not connectivity preserving. It should be replaced by some skeleton routines similar to WATERSHED. This comprises a skeleton, an exoskeleton, a skeleton by zones of influence (SKIZ) etc. in fact the implementation of MORPH_THIN/THICK in a loop. For large images a mask parameter restricting on the mask pixels could be helpful (also for watershed).

LABEL_REGION
Although not directly MM related it is one of the jewels of IDL (like where and histogram). The disturbing thing is (again) the border point behavior. It is deleting a one bit border. This is understandable at times where memory restrictions were severe but today?!?. Why not replacing the border points set by the newly calculated label values? Using LABEL_REGION e.g. for filling up objects (labeling the complement, deleting the background object, (res gt 0) OR with original) would become an easy task!

WATERSHED
First: why not implementing a 3d version of it. Only the choice of the neighborhood (as pointer array in the Vincent algorithm) should be changed, very easy and very helpful.
Second: WATERSHED is a labelling algorithm, it should correspond with LABEL_REGION! I am using watershed on binary images where the border behavior of LABEL_REGION is disturbing. Connectivity keyword corresponds to st.e.!

Am Thu, 12 Oct 2006 16:56:26 +0200 schrieb Karl Schultz <k_remove_schultz@ittvis.com>:

> On Thu, 12 Oct 2006 09:41:29 +0200, Karsten Rodenacker wrote:
>
>> Don't use IDL's dilate and erode without embedding your data into a
>> sufficiently large array. Border handling is not coherently implemented.
>> That is a large disadvantage, not to say an error, for the application
>> of
>> math. morph. operations in sequences. Ask for improvement, possibly
>> ITTVIS
>> can be convinced!
>> Regards
>> Karsten
>>
>> Am Thu, 12 Oct 2006 04:33:59 +0200 schrieb Gongqin Shen
>> <gqshen2008@gmail.com>:
>>
>>>> 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?
>>>
>
> All I can say is that we know about this problem and fixing it is "on the
> list". Karsten has already sent me some more detail. If anyone else
> would like to submit additional input, besides what is already in this
> thread, email it to kschultz at ittvis dot com. Thanks!
>
> Karl

--
Erstellt mit Operas revolutionï¿œrem E-Mail-Modul: http://www.opera.com/m2/