Subject: Re: Problems with ERODE and DILATE functions
Posted by on Wed, 17 Nov 2010 07:57:50 GMT
View Forum Message <> Reply to Message

On 16 nov, 22:44, James <donje...@gmail.com> wrote:
> On Nov 16, 8:02 am, Oriol Güell Riera <oriolguellri...@gmail.com>
> wrote:
>
>
>
>> On 16 nov, 16:56, David Fanning <n...@dfanning.com> wrote:
>
>>> Oriol Güell Riera writes:
>>>> The problem is that it works fine for small stacks of images but not
>>>> for large ones, the pc crashes.
>
>>> Yes, and I presume you think it should work for large
>>> stacks of images. That's the assumption I think I would
>>> focus the investigation on. You assume large stacks
>>> are similar to small stacks in all ways except size.
>>> Are they? Does your program evolve over time to do
>>> something you don't expect it to? Larger stacks
>>> mean longer running times. These are just two of
>>> the hundred or so theories I could come up with if
>>> I had five minutes. You are going to have to play
>>> detective this morning. Don't you watch CSI? It's
>>> going to be FUN! :-)
>
>>> Cheers,
>
>>> David
>
>>> --
>>> David Fanning, Ph.D.
>>> Fanning Software Consulting, Inc.
>>> Coyote's Guide to IDL Programming:http://www.dfanning.com/
>>> Sepore ma de ni thui. ("Perhaps thou speakest truth.")
>
>> It seems I will have to do become Sherlock Holmes, because when I use
>> erode and dilate for the large stack it works fine, the problem is
>> that it detects more particles because erode and dilate separate the
>> pixels. Due to this, I erase erode and dilate, so I think that the
>> only modification in the program is that the pixels are not going to
>> get modified, but the rest of the program is exactly the same. The
>> thing that makes me go mad is that the program works with erode and
>> dilate and it doesn't work when I remove them. It is strange, it
>> should work without modifying the image.
>> I'll keep investigating.

>> Thank you again David,
>> Oriol
>
> Can you post the code for the program you're working on? As David
> says, it sounds like there is some unintended behavior in the program
> that you're not accounting for.
>
> One thing to consider: DILATE and ERODE are converting your data to
> Byte type unless you are using both the /GREY and /PRESERVE_TYPE
> keywords. Perhaps your input data is in a larger type, and when you
> remove the DILATE/ERODE calls, the data is too large for later stages
> of processing.

Here's the routine. The main program calls it in some point to find
the particles and compute the center of mass position and the
orientation:

```
---------------------------------------------
function tracking,cube,num,result

result=cube

nx=n_elements(cube(*,0,0))
ny=n_elements(cube(0,*,0))
nz=n_elements(cube(0,0,*))
bloblist=fltarr(4)
blank=bytarr(nx,ny)
the=fltarr(num*nz)
k=-1
oa=0

for j=0,nz-1 do begin
        o=0
        if j eq oa*10 then begin
                message,'slice '+string(j)+' of '+string(nz-1),/inf
                oa=oa+1
        endif
        slice=result(*,*,j)

        ;The next 5 lines are the ones that I'm trying to erase, but
when I put them off the program crashes!
        lilsquare=bytarr(3,3)+1b
        bigsquare=bytarr(5,5)+1b
        ; next three lines do a good job removing tiny islands and
craters
        slice=erode(slice,lilsquare)
        slice=dilate(slice,bigsquare)
        slice=erode(slice,lilsquare)
```

```
    s=size(slice)
    w=where(slice eq 1,nw)
    if (nw gt 0) then begin
        while (nw gt 0) do begin
            ypos2=w(0)/(s[1])
            xpos2=w(0)-(ypos2*s[1])
            region=search2d(slice,xpos2,ypos2,1,1)
            if (n_elements(region) gt 1) then begin
                k=k+1
                o=o+1
                ypos=region/(s[1])
                xpos=(region-(ypos*s[1]))
                numpts=n_elements(region)
                mass=numpts
                massxy=slice(xpos,ypos)
                xbar=total(xpos*massxy)/mass
                ybar=total(ypos*massxy)/mass
                rx=xpos-xbar
                ry=ypos-ybar
                i11=total(rx*rx)/mass
                i22=total(ry*ry)/mass
                i12=total(rx*ry)/mass
                trace=i11+i22
                det=i11*i22-i12*i12
                eval1=trace/2+sqrt(trace^2/4-det)
                xeix1=1./sqrt(1.+((eval1-i11)/i12)^2)
                yeix1=(eval1-i11)/i12*1./sqrt(1.+
((eval1-i11)/i12)^2)
                if i12 eq 0 then begin
                    xeix1=1.
                    yeix1=0.
                endif
                if (j eq 0) then begin
                    the(k)=!pi-atan2(yeix1,xeix1)
                endif else begin
                    theta1=!pi-atan2(yeix1,xeix1)
                    theta2=!pi-atan2(-yeix1,-
xeix1)
                    if (abs(theta1-the(k-num)) gt
2) then begin
                        the(k)=theta2
                    endif else begin
                        the(k)=theta1
                    endelse
                endelse
                ybar=float(ny)-temporary(ybar)
```

```
                    bloblist = [[bloblist],
[xbar,ybar,the(k),j]]
                    if o gt num then begin
                         print,'Slice',j,' has a
problem'
                    endif
                    slice(region) = 255b
                    w=where(slice eq 1,nw)
               endif
          end
     endif else begin
          message,'WARNING: no pixels above threshold',/inf
     endelse
     result(*,*,j)=slice
endfor
; strips off first empty row
bloblist=bloblist(*,1:*)

return, bloblist

end
```

The input images are thresholded, they only have 0 and 1. Do you suggest to write slice=byte(temporary(slice)) instead of erode and dilate to transform it to byte type?

Thanks again David and James
Oriol