
Subject: Re: filling holes in blobs quickly

Posted by [Karsten Rodenacker](#) on Thu, 18 Apr 2002 17:07:12 GMT

[View Forum Message](#) <> [Reply to Message](#)

Oups, I forgot function mm_bord. It creates an array of same size with borders two bits set (with keyword /del) .

```
;+
;
NAME:
;
MM_BORD
;
function mm_bord,a,n,VALUE=val,Z_FIRST=z_fir, ALL=all, DELETE = del
if n_params() eq 1 then n=1
if not keyword_set(val) then val=0
sx=size(a)
IF NOT keyword_set(del) THEN e=a ELSE e = make_array(size = sx)
case sx(0) of
1:begin
  e([0,sx(1)-1])=val
  end
2:begin
  wx=[indgen(n),sx(1)-indgen(n)-1]
  wy=[indgen(n),sx(2)-indgen(n)-1]
  e(wx,*)=val & e(*,wy)=val
  end
3:begin
  if keyword_set(z_fir) then begin
    wx=[indgen(n),sx(2)-indgen(n)-1]
    wy=[indgen(n),sx(3)-indgen(n)-1]
    e(*,wx,*)=val & e(*,*,wy)=val
    if keyword_set(all) then begin
      e[0:n-1,*,*]=val
      e[sx[1]-n:sx[1]-1,*,*]=val
    endif
  endif else begin
    wx=[indgen(n),sx(1)-indgen(n)-1]
    wy=[indgen(n),sx(2)-indgen(n)-1]
    e(wx,*,*)=val & e(*,wy,*)=val
    if keyword_set(all) then begin
      e[*,*,0:n-1]=val
      e[*,*,sx[3]-n:sx[3]-1]=val
    endif
  endelse
end
4:begin ; first coordinate not considered
  if keyword_set(z_fir) then begin
    wx=[indgen(n),sx(3)-indgen(n)-1]
```

```

wy=[indgen(n),sx(4)-indgen(n)-1]
e(*,*,wx,*)=val & e(*,*,*,wy)=val
if keyword_set(all) then begin
  e[*],0:n-1,*,*]=val
  e[*],sx[1]-n:sx[1]-1,*,*]=val
endif
endif else begin
  wx=[indgen(n),sx(2)-indgen(n)-1]
  wy=[indgen(n),sx(3)-indgen(n)-1]
  e(*,wx,*,*)=val & e(*,*,*,wy)=val
  if keyword_set(all) then begin
    e[*],*,*,0:n-1]=val
    e[*],*,*,sx[1]-n:sx[1]-1]=val
  endif
endif
endelse
end
else:
endcase
return,e
end
;-

```

May be this routine does what you like. However, there should be background touching the image border!

Regards

```

;+
; NAME:
;
MM_FILL_NEW
;
;
; PURPOSE:
;
Fills holes
;
;
; CATEGORY:
;
Image processing
;
;
; CALLING SEQUENCE:
;
result = MM_FILL_NEW(mask)
;
;
; INPUTS:

```

```

;
;
mask:
An image from which only pixels ne 0 are considered
;
;
;
; OPTIONAL INPUTS:
;
;
none
;
;
; KEYWORD PARAMETERS:
;
;
EIGHT:
Neighborhood used is 8-connected (def. 4)
;
; see label_region
;
;
; OUTPUTS:
;
; A filled byte mask image
;
;
; OPTIONAL OUTPUTS:
;
;
none
;
;
; COMMON BLOCKS:
;
;
none
;
;
; SIDE EFFECTS:
;
; label_region is used
;
;
; RESTRICTIONS:
;
;
;
;
; PROCEDURE:
;
;
;
;
; EXAMPLE:
;
;

```

```

;
;
;
; MODIFICATION HISTORY:
;
;   Wed Oct 1 09:11:34 1997, Karsten Rodenacker
;   <rodena@gsf.de>
;
;
;
;
;-
FUNCTION MM_FILL_NEW, inp, ALL = all, _EXTRA=ext
  b = byte(inp NE 0)
  l = label_region(1b-b, _EXTRA=ext)
  bg = where(histogram(byte(l*mm_bord(b,2,val=1, /del, all = all))) ne
0, cbg)
  hl = uindgen(max(l)+1)
  hl[bg] = 0
  return, b OR (hl[l] NE 0)
; print,cbg, bg
; IF cbg GT 1 THEN FOR i=1, cbg-1 DO BEGIN
;   l[where(l EQ bg[i])] = 0
;   tvscl, l
; ENDFOR
; return, b OR byte(l NE 0)
end

```

Doug Martin wrote:

```

> I'm sure this is a well solved problem, but:
> I'm trying to fill in holes inside of domains (blobs) quickly. The
> domains are in a binary image, and are bright.
>
> Right now, I search for dark regions - the holes (label_region);
> pick out those of a small size (histogram and where);
> and then fill them in (a FOR loop through the where values, since I
> don't know how to get where to compare to every value in a vector).
>
> This is slow, and I strongly suspect there is an intelligent way to do
> this: fill in all dark blobs up to a certain size.
>
> Thanks for any help,
>
> Doug Martin
> University of Texas at Austin
>

```

--

Karsten Rodenacker ()

-----:-)

GSF - Forschungszentrum Institute of Biomathematics and Biometry

D-85758 Oberschleissheim Postfach 11 29

Tel: +49 89 31873401 | FAX: ...3369 | rodена@gsf.de |

Karsten@Rodenacker.de

<http://www.gsf.de/ibb/homepages/rodenacker>
