

---

Subject: Re: MORPH\_XYZ and values keyword  
Posted by [Karsten Rodenacker](#) on Wed, 25 Jun 2003 07:10:21 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hallo, maybe some information concerning VALUE keywords. Let ste the structuring element and val the values array.

1. Only val\*(ste ne 0) is of importance!
2. As far as I know are the val values subtracted or added from the input data before the max or min according erode or dilate is performed.

Hence, flat value arrays do not make sense.

3. I attach two routines (for 1-d morphological filtering) for illustration. gen\_sphere to generate the value array and mm\_filter to apply a sequence of increasing filter steps. Unluckily the routines are not documented. Maybe they are comprehensible without further words.

Regards  
Karsten

Ben Tupper schrieb:

- > Hello,
- >
- > I completely mystified by the VALUES keyword to the MORPH\_XYZ (like
- > MORPH\_TOPHAT) routines. I think I understand the meaning of VALUES for
- > DILATE and ERODE-like routines - but I can't make sense of the MORPH\_XYZ
- > routines.
- >
- > Here's the description of the so-called structuring element that these
- > routine accept...it gives the shape and 'on-off'-ness of each pixel.
- >
- > A one-, two-, or three-dimensional array to be used as the structuring
- > element. The elements are interpreted as binary values  $\bar{i} \in \{0, 1\}$  either zero or
- > nonzero. The structuring element must have the same number of dimensions
- > as the Image argument.
- >
- >
- >
- > Now here's the decription of the VALUES keyword...
- >
- > An array of the same dimensions as the Structure argument providing the
- > values of the structuring element. If the VALUES keyword is not present,
- > all elements of the structuring element are 0.
- >
- >
- > What the heck does that mean?
- >
- > I thought providing the VALUES was like
- > providing a weighting to the structuring element.
- >

>  
> Here's an example using the MORPH\_TOPHAT routine... if here's a  
> structure with a narrow brim - any peaks detected that fit within the 1s  
> is a keeper.

```
>  
> 0 0 0 1 0 0 0  
> 0 1 1 1 1 1 0  
> 0 1 1 1 1 1 0  
> 1 1 1 1 1 1 1  
> 0 1 1 1 1 1 0  
> 0 1 1 1 1 1 0  
> 0 0 0 1 0 0 0
```

>  
> Now if I am working with a grayscale image... I want to make sure the  
> peak is tall (bright) enough so, I weight the structuring element with  
> some kind of a height - like I want the peak to be at least 5 whatever  
> above the brim values. So I set VALUES equal to...

```
>  
> 0 0 0 5 0 0 0  
> 0 5 5 5 5 5 0  
> 0 5 5 5 5 5 0  
> 5 5 5 5 5 5 5  
> 0 5 5 5 5 5 0  
> 0 5 5 5 5 5 0  
> 0 0 0 5 0 0 0
```

>  
>  
> I can't see any difference between the results when I do and do not  
> provide the VALUES. Which, of course, means I don't understand it at all.

>  
> Stumped and grateful for any help,

>  
> Ben

>

```
function mm_filter,vt,start,ende,LINEAR=linear,ST_TYPE=typ,FIRST=firs t  
;  
; Ouverture und Fermeture nach Sternberg (Morphological filters ?)  
;  
if keyword_set(linear) then stern=0 else stern=1  
if not keyword_set(typ) then typ=0 ;dreieck  
if not keyword_set(first) then first=0 ;ouverture am Anfang  
svt=size(vt)  
xvt=[vt[svt[1]-indgen(svt[1])-1],vt,vt[svt[1]-indgen(svt[1]) -1]]  
low=xvt  
upp=xvt  
for l=start,ende do begin
```

```

kerg=gen_sphere(l,typ=typ)
if not first then begin
  if stern then begin
    low=dilate(erode(low,kerg ge 0,val=kerg),kerg gt 0,val=kerg)
    upp=erode(dilate(upp,kerg ge 0,val=kerg),kerg gt 0,val=kerg)
  endif else begin
    low=dilate(erode(low,kerg ge 0,/gr),kerg ge 0,/gr)
    upp=erode(dilate(upp,kerg ge 0,/gr),kerg ge 0,/gr)
  endelse
endif else begin
  if stern then begin
    low=erode(dilate(low,kerg ge 0,val=kerg),kerg ge 0,val=kerg)
    upp=dilate(erode(upp,kerg ge 0,val=kerg),kerg ge 0,val=kerg)
  endif else begin
    low=erode(dilate(low,kerg ge 0,/gr),kerg ge 0,/gr)
    upp=dilate(erode(upp,kerg ge 0,/gr),kerg ge 0,/gr)
  endelse
endelse
first=(not first) and 1
endfor
return,[[upp[svt[1]:svt[1]*2-1]],[[low[svt[1]:svt[1]*2-1]]]
end
res=mm_filter(vt4,1,4,fir=0,st_typ=1)&plot,vt4,col=1,psy
m=5,syms=0.25&oplot,res[*],col=2&oplot,res[*],1,co l=2&wshow,1
end
function gen_sphere, rad, TYPE=typ

```

```

if not keyword_set(typ) then typ=0
case typ of
  1:v = sqrt(rad^2-(rad-findgen(rad*2+1))^2)
  2:v = rad^2-(rad-findgen(rad*2+1))^2
  3:v = 0.05/(1.+sqrt(rad^2-(rad-findgen(rad*2+1))^2))
  4:v = 0.05/(1.+rad^2-(rad-findgen(rad*2+1))^2)
  5:v = rad-sqrt(rad^2-(rad-findgen(rad*2+1))^2)
  6:v = rad^2-(rad-findgen(rad*2+1))^2
  -1:v = replicate(1.0,rad*2+1)
else: v = rad-abs(-findgen(rad*2+1)+rad)
ENDCASE
mv = max(v)
IF mv GT (rad+1) THEN v = v/mv*(rad+1)
return, v
end

```

## File Attachments

- 1) [mm\\_filter.pro](#), downloaded 120 times
- 2) [gen\\_sphere.pro](#), downloaded 109 times