Subject: Re: Local max filter
Posted by Craig Markwardt on Tue, 21 Aug 2001 21:09:16 GMT
View Forum Message <> Reply to Message

```
rkj@dukebar.crml.uab.edu (R. Kyle Justice) writes:

> I am trying to implement a local max filter

> without loops. Has this been done?

> (Given an array and a filter width, return an

> array containing the array value if it is

> a local max, 0 if not)

> For instance,

> 3 4 7 2 6 4 9 8 3

> would be

> 0 0 7 0 0 0 9 0 0

> for a width of 5.
```

JD and I had a contest doing this kind of thing -- finding maxima -- a year or so ago. Of course I popped his socks off, but he will tell you a different story :-)

Perhaps the easiest way to do this is with a bunch of vector compares.

```
arr = [3, 4, 7, 2, 6, 4, 9, 8, 3] ;; Initial data

arr2 = arr(2:*) ;; Center points

b = (arr2 GE arr(0:*)) AND (arr2 GE arr(1:*)) AND $

(arr2 GE arr(3:*)) AND (arr2 GE arr(4:*)) ;; Compare against neighbors

result = [0, 0, b*arr2, 0, 0] ;; Replace boundaries
```

The trick is that you are comparing arr(2:*) to each of its neighbors. I am using the little trick I've published a couple of times, which is that when two vectors of unequal lengths are compared, the longer one is truncated to the other one's size. Otherwise you need to do "arr2 GE arr(0:n_elements(arr)-2)" and so on.

If you really need variable widths then the above can be formulated into a loop over the width. This is not a hurtful loop because the arrays are still compared vectorially. Try this function out:

```
function locmax, arr, width
if n_elements(arr) LT width then message, 'ERROR: arr is too small'
if (width MOD 2) EQ 0 then message, 'ERROR: width must be odd'
ic = (width-1)/2
```

```
arrc = arr(ic:*)
 b = bytarr(n elements(arr)-width+1) + 1b
 for i = 1, ic do $
   b = b AND (arrc GE arr(ic-i:*)) AND (arrc GE arr(ic+i:*))
 return, [arr(0:ic-1)*0, b*arrc, arr(0:ic-1)*0]
end
I play the same tricks, plus a few tricks to preserve the type of the
original array.
Craig
                                EMAIL: craigmnet@cow.physics.wisc.edu
Craig B. Markwardt, Ph.D.
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
Subject: Re: Local max filter
Posted by rkj on Tue, 21 Aug 2001 21:55:11 GMT
View Forum Message <> Reply to Message
Thanks!
I was afraid any potential solution would involve "histogram" ;-)
As an aside, would this work in Matlab? Some people around here
just won't make the switch . . .
Kyle
Craig Markwardt (craigmnet@cow.physics.wisc.edu) wrote:
: rkj@dukebar.crml.uab.edu (R. Kyle Justice) writes:
: > I am trying to implement a local max filter
: > without loops. Has this been done?
: >
: > (Given an array and a filter width, return an
: > array containing the array value if it is
```

: > a local max, 0 if not)

: > For instance,

:>

:>347264983

```
: > would be
:>007000900
: > for a width of 5.
: JD and I had a contest doing this kind of thing -- finding maxima -- a
: year or so ago. Of course I popped his socks off, but he will tell
: you a different story :-)
: Perhaps the easiest way to do this is with a bunch of vector compares.
: arr = [3, 4, 7, 2, 6, 4, 9, 8, 3] ;; Initial data
: arr2 = arr(2:*)
                            ;; Center points
: b = (arr2 GE arr(0:*)) AND (arr2 GE arr(1:*)) AND $
   (arr2 GE arr(3:*)) AND (arr2 GE arr(4:*)) ;; Compare against neighbors
: result = [0, 0, b*arr2, 0, 0] ;; Replace boundaries
: The trick is that you are comparing arr(2:*) to each of its neighbors.
: I am using the little trick I've published a couple of times, which is
: that when two vectors of unequal lengths are compared, the longer one
: is truncated to the other one's size. Otherwise you need to do "arr2
: GE arr(0:n_elements(arr)-2)" and so on.
: If you really need variable widths then the above can be formulated
: into a loop over the width. This is not a hurtful loop because the
: arrays are still compared vectorially. Try this function out:
: function locmax, arr, width
: if n_elements(arr) LT width then message, 'ERROR: arr is too small'
: if (width MOD 2) EQ 0 then message, 'ERROR: width must be odd'
: ic = (width-1)/2
: arrc = arr(ic:*)
: b = bytarr(n_elements(arr)-width+1) + 1b
: for i = 1, ic do $
    b = b AND (arrc GE arr(ic-i:*)) AND (arrc GE arr(ic+i:*))
 return, [arr(0:ic-1)*0, b*arrc, arr(0:ic-1)*0]
: end
: I play the same tricks, plus a few tricks to preserve the type of the
: original array.
: Craig
: Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
: Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
```

Subject: Re: Local max filter

Posted by air_jlin on Tue, 21 Aug 2001 22:57:50 GMT

View Forum Message <> Reply to Message

hi Kyle,

i think what you're looking for is in Ray Sterner's Johns Hopkins library (extremes.pro and extremes_2d.pro), although in those routines the filter width is fixed. below is the link to Martin Schultz's web listing of the library:

http://www.mpimet.mpg.de/~schultz.martin/idl/html/libjhuapls 1r.html

best,
-Johnny

Johnny Lin
CIRES, University of Colorado
Work Phone: (303) 735-1636
Web: http://cires.colorado.edu/~johnny/

.....

rkj@dukebar.crml.uab.edu (R. Kyle Justice) wrote in message news:<9lu862\$ci1\$1@SonOfMaze.dpo.uab.edu>...

- > I am trying to implement a local max filter
- > without loops. Has this been done?

> (Given an array and a filter width, return an

- > array containing the array value if it is
- > a local max, 0 if not)

> For instance,

>

> 347264983

> would be

> 007000900

> for a width of 5.

> Kyle J.

View Forum Message <> Reply to Message

```
Craig Markwardt wrote:
> rkj@dukebar.crml.uab.edu (R. Kyle Justice) writes:
>> I am trying to implement a local max filter
>> without loops. Has this been done?
>>
>> (Given an array and a filter width, return an
>> array containing the array value if it is
    a local max, 0 if not)
>>
>> For instance,
>>
>> 347264983
>> would be
>>
>> 007000900
>>
>> for a width of 5.
> JD and I had a contest doing this kind of thing -- finding maxima -- a
> year or so ago. Of course I popped his socks off, but he will tell
> you a different story :-)
Hmmmph... from my posting before of April 2000:
"Nice entry Craig. But unfortunatly it doesn't *alway* do exactly what
requested. It works fine for n=5, but for n>5 (7,9,...), the index is
off..."
So it looks like your method owes me at least a bit of bug fixing;)
This is the thread for those interested:
http://groups.google.com/groups?hl=en&safe=off&th=f2 0f62ee42b51402,20&start=0
The bottom line of my method was, for simple 3 pt maxima:
maxes = where(arr gt median(arr,3))
<<<<
which I guess when compared with Craig's:
```

```
>>>> >>
arr2 = arr(2:*)
                          ;; Center points
b = (arr2 GE arr(0:*)) AND (arr2 GE arr(1:*)) AND $
  (arr2 GE arr(3:*)) AND (arr2 GE arr(4:*)) ;; Compare against
neighbors
result = [0, 0, b*arr2, 0, 0] ;; Replace boundaries
<<<<<
```

does get its socked knocked off, at least in terms of amount of typing required, space used on disk, or impressive complex subscripts to more quickly glaze the boss's eyes over;)

As for variable width, n-point maxima (n odd), I came up with:

wh=where(d gt ((m=median(d,3))) and smooth((d eq m)*(n-2),n-2) eq n-3)

Not a for loop (or a histogram) in there! For the interested, the details of this technique were well described (a.k.a. "How the hell does that work?").

JD

P.S. Craig, now that I have your attention, I have an unrelated question, the answer to which might be of general interest. How do you feel about your excellent fitting/minimization routines being distributed with a large scale freely available system for scientific reduction and analysis?

Subject: Re: Local max filter Posted by Craig Markwardt on Wed, 22 Aug 2001 22:48:40 GMT View Forum Message <> Reply to Message

JD Smith <idsmith@astro.cornell.edu> writes:

- >> JD and I had a contest doing this kind of thing -- finding maxima -- a
- >> year or so ago. Of course I popped his socks off, but he will tell
- >> you a different story :-)

> Hmmmph... from my posting before of April 2000:

- "Nice entry Craig. But unfortunatly it doesn't *alway* do exactly what
- > requested. It works fine for n=5, but for n>5 (7,9,...), the index is
- > off..."

Yeah, I was just baiting you a little. :-)

- > P.S. Craig, now that I have your attention, I have an unrelated
- > question, the answer to which might be of general interest. How do you
- > feel about your excellent fitting/minimization routines being
- > distributed with a large scale freely available system for scientific
- > reduction and analysis?

I feel fine about that. Of course, someone *else* will provide the support, not me. :-) Any more details?

Craig

Subject: Re: Local max filter
Posted by Martin Downing on Tue, 28 Aug 2001 09:47:29 GMT
View Forum Message <> Reply to Message

Hi All,

I almost have a solution for N dimensional arrays, but since it is a SHIFT based solution I need help on solving the wrap problem (well its a problem to me!) -

i.e.. I need to prevent wrap and instead replicate the last elements of the array in that dimension: e.g.: what we have is:

```
> print, shift([-2,1,3,8],1)
8 -2 1 3
```

we would like

```
> print, shift([-2,1,3,8],1, /REPLICATE_BORDER)
-2 -2 1 3
```

So in trying to solve this problem I am laying down another challenge for the boys;)

Anyway - heres my code for those interested - its still ok for large arrays and could always be padded round the edges.

function local_maxima, array, width, INDEX=INDEX
;+
; Purpose: Calculates local maxima of ARRAY of any dimension
; Restrictions: incorrect results around the array border of WIDTH elements
; maybe be produced due to maxima wrapping from use of shift
; allows multiple maxima within WIDTH if identical
; - should be fairly fast even for large WIDTHs
; MRD 26/08/2001
;marr=array ; temp array in which maxima "dilate"
s = size(marr)
ndims = s[0]

```
sarr = intarr(ndims) ; array for controlling shift
sarr[0] = 1 ; set to shift one dimension at a time
for w = 1, width do begin
for d = 1, ndims do begin
 marr = shift(marr, sarr) > shift(marr, -sarr) > marr
 ; NOTE need shift to replicate border pixels rather than wrapping
 sarr = shift(sarr, 1)
endfor
endfor
if keyword set(INDEX) then begin
return, where (marr eq array)
endif else begin
return, (marr eq array)*marr
endelse
end
; END OF LOCAL MAXIMA
Martin Downing,
Orthopaedic RSA Research Centre,
Woodend Hospital, Aberdeen, AB15 6LS.
m.downing@abdn.ac.uk
"Craig Markwardt" <craigmnet@cow.physics.wisc.edu> wrote in message
news:onn14ryalz.fsf@cow.physics.wisc.edu...
> JD Smith <idsmith@astro.cornell.edu> writes:
>>> JD and I had a contest doing this kind of thing -- finding maxima -- a
>>> year or so ago. Of course I popped his socks off, but he will tell
>>> you a different story :-)
>>
>> Hmmmph... from my posting before of April 2000:
>> "Nice entry Craig. But unfortunatly it doesn't *alway* do exactly what
>> requested. It works fine for n=5, but for n>5 (7,9,...), the index is
>> off..."
> Yeah, I was just baiting you a little. :-)
>
>
>> P.S. Craig, now that I have your attention, I have an unrelated
>> guestion, the answer to which might be of general interest. How do you
```

```
>> feel about your excellent fitting/minimization routines being
>> distributed with a large scale freely available system for scientific
>> reduction and analysis?
>
> I feel fine about that. Of course, someone *else* will provide the
> support, not me. :-) Any more details?
>
> Craig
```