Subject: Re: find bimodal maximum in each row
Posted by Jeremy Bailin on Fri, 04 Feb 2011 20:57:00 GMT
View Forum Message <> Reply to Message

On Friday, February 4, 2011 4:02:47 AM UTC-5, vijay wrote:
> I have an array of 512x512 and for each row there is a
> bimodal like peak. I want to find this peak in each row (ex. first
> peak value 177 and second peak value 244, etc). Like that each row
> will have two different peak
> values. How to obtain these peak values for the whole array and result
> them in new array.

If you can guarantee that the peaks will not occur in the first or last column of each row, this will give you every single local maximum:

```
q = [[0,1,2,1,2,1,0], [1,2,0,0,2,2,0], [0,2,2,2,0,1,0]]
nx = (size(q, /dimen))[0]

; is this a local maximum?  (note that the index starts at element 1,* of q)
peakp = (q[1:nx-2,*] gt q[2:nx-1,*]) and (q[0:nx-3,*] le q[1:nx-2,*])
; where is that true?
peaklocations = where(peakp)
; turn into array locations into peakp
peaklocations_xy = array_indices(peakp, peaklocations)
; increment the x coordinate because peakp is shifted one element to the right of q
peaklocations[0,*]++
; what are the values of q there?
peakvals = q[peaklocations_xy[0,*], peaklocations_xy[1,*]]

IDL> print, peaklocations_xy
       2       0
       4       0
       1       1
       5       1
       3       2
       5       2
IDL> print, peakvals
      2
      2
      2
      2
      2
      1
```

Note that it will pick up every single local maximum, regardless of how many there are per row.

-Jeremy.

Subject: Re: find bimodal maximum in each row
Posted by vijay on Sat, 05 Feb 2011 03:32:40 GMT
View Forum Message <> Reply to Message

On Feb 5, 1:57 am, Jeremy Bailin <astroco...@gmail.com> wrote:
> On Friday, February 4, 2011 4:02:47 AM UTC-5, vijay wrote:
>> I have an array of 512x512 and for each row there is a
>> bimodal like peak. I want to find this peak in each row (ex. first
>> peak value 177 and second peak value 244, etc). Like that each row
>> will have two different peak
>> values. How to obtain these peak values for the whole array and result
>> them in new array.
>
> If you can guarantee that the peaks will not occur in the first or last column of each row, this will give you every single local maximum:
>
> q = [[0,1,2,1,2,1,0], [1,2,0,0,2,2,0], [0,2,2,2,0,1,0]]
> nx = (size(q, /dimen))[0]
>
> ; is this a local maximum? (note that the index starts at element 1,* of q)
> peakp = (q[1:nx-2,*] gt q[2:nx-1,*]) and (q[0:nx-3,*] le q[1:nx-2,*])
> ; where is that true?
> peaklocations = where(peakp)
> ; turn into array locations into peakp
> peaklocations_xy = array_indices(peakp, peaklocations)
> ; increment the x coordinate because peakp is shifted one element to the right of q
> peaklocations[0,*]++
> ; what are the values of q there?
> peakvals = q[peaklocations_xy[0,*], peaklocations_xy[1,*]]
>
> IDL> print, peaklocations_xy
> 2 0
> 4 0
> 1 1
> 5 1
> 3 2
> 5 2
> IDL> print, peakvals
> 2
> 2
> 2
> 2
> 2
> 1
>
> Note that it will pick up every single local maximum, regardless of how many there are per row.
>
> -Jeremy.

hi jeremy,

that will give the peak values in one dimensional, but i want the peak values in an array as the same size of input array and non-peak tends to be zero. Thus i will have an array (bcos i am finding peak in image in each row).

Subject: Re: find bimodal maximum in each row
Posted by Jeremy Bailin on Sun, 06 Feb 2011 20:27:45 GMT
View Forum Message <> Reply to Message

That's easily constructed from peakp:

```
q = [[0,1,2,1,2,1,0], [1,2,0,0,2,2,0], [0,2,2,2,0,1,0]]
nx = (size(q, /dimen))[0]
ny = (size(q, /dimen))[1]

; is this a local maximum?  (note that the index starts at element 1,* of q)
peakp = (q[1:nx-2,*] gt q[2:nx-1,*]) and (q[0:nx-3,*] le q[1:nx-2,*])
; pad peakp to make it the same dimensions as q and multiply
peakonly = q * [replicate(0,1,ny), peakp, replicate(0,1,ny)]


IDL> print, peakonly
       0       0       2       0       2       0       0
       0       2       0       0       0       2       0
       0       0       0       2       0       1       0
```

-Jeremy.
