
Subject: Re: not-quite median filter

Posted by [jeanh](#) on Thu, 30 Sep 2010 20:51:42 GMT

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

On 30/09/2010 3:38 PM, JJ wrote:

> I would like to do something that is similar to, but not quite the
> same as, a median filter to a 2D array. Instead of choosing the
> median value in a box surrounding each pixel, I would like to choose
> the value in that box that occurs most frequently.

>

> For example, if I had

>

> 1 1 1 1 1

> 1 1 1 1 1

> 2 2 2 2 2

> 2 2 3 3 3

> 3 3 3 3 3

>

> The median would be 2, but I would want the value 1 (it occurs 10
> times, which is more than the 8 instances of the value 3 or the 7
> instances of the value 2).

>

> Can anyone think of a clever way to do this that would be fast in IDL
> (ie, no looping through the pixels)? I need it to work for box sizes
> up to around 21. Ties may be broken arbitrarily.

>

> Is there already a name for this concept?

>

> Thanks.

>

> -Jonathan

Hi,
this is what I use (similar to Gray's suggestion)

```
function getMajority,data, central=central
;This function returns the majority value of the input array.
;If there is a tie, a random mode is returned.
;If the central keyword is specified, if there is a tie and the value of
the central cell of the array is in one of
;the mode, then this value is returned.
;
;
;Written by Jean-Gabriel Hasbani (firstname@lastname.ca)
;December 2007
```

```

majority = -1
;Use histogram to find the mode
histo = histogram(data, min=0, r=ri)
;Get the number of time the mode(s) value is repeated
MaxVal = max(histo)
;Find every entry in the histogram that have this frequency
maxValInd = where(histo eq maxVal, count)

;If there is only 1 mode, return its value
if count eq 1 then return, data[ri[ri[maxValInd]]]

;else, look at all the possible values (modes) and return a random one

;If the keyword is specified, if the central cell is of the type of one
of the majority, return this value
if keyword_set(central) then begin
  ;get the central cell value
  centralValue = data[n_elements(data)/2 - 1]
  ;For each mode, get the value and compare is with the central value
  for i = 0, count-1 do begin
    histoData = data[ri[ri[maxValInd[i]]]]
    if histoData eq centralValue then return, centralValue
  endfor
endif

;Create a random index
randomID = randomu(seed,count)
randomID = (sort(randomID))[0]

;return the random mode value
return, data[ri[ri[maxValInd[randomID]]]]
end

```

Jean
