

---

Subject: Re: not-quite median filter

Posted by [Jeremy Bailin](#) on Fri, 01 Oct 2010 11:42:56 GMT

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

---

On Sep 30, 3:38 pm, JJ <j...@cornell.edu> 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

Any non-loop solution I've thought of is going to require a lot of memory, since you're going to end up storing one slice for each central pixel. For images of size no more than 21, that might be okay, but it won't work much larger. I'd do something like this:

- create an array with each filter-box-over-an-individual-pixel as a separate 2D slice in a 3D array
- use `VALUE_LOCATE` to map your values onto simple integers from 0 to N-1
- increment the values in slice #i by  $i*N$  so that the values in each slice are unique
- perform the histogram, for which entries  $i*N$  through  $(i+1)*N-1$  are the repeat counts for slice i
- reform the histogram into a 2D array, N by n\_slices, and run `SORT_ND` on it so that you know where the maximum histogram value for each slice is

- subtract back out  $i*N$  from that value, reform back to the original dimensions, and do the reverse mapping to go back to your values

-Jeremy.

---