Subject: Re: Non maxima supression
Posted by David Fanning on Mon, 27 Nov 2006 19:09:24 GMT
View Forum Message <> Reply to Message

Charudatta Phatak writes:

> I want to do a non maxima supression on a 2d array within a specified
> neighborhood of pixels. The way i am doing it right now is looping over
> all the pixels and check if the value is max in the 5x5 neighborhood. if
> it is max then keep it or else set it to zero. Is there a IDL way to do
> it faster than 2 for loops?

Can you use any of the lessons learned in this "maximum value
array resampling" article:

http//www.dfanning.com/idl_way/maxresample.html

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Non maxima supression
Posted by Jean H. on Mon, 27 Nov 2006 21:01:02 GMT
View Forum Message <> Reply to Message

Charudatta Phatak wrote:
> Hello,
>
> I want to do a non maxima supression on a 2d array within a specified
> neighborhood of pixels. The way i am doing it right now is looping over
> all the pixels and check if the value is max in the 5x5 neighborhood. if
> it is max then keep it or else set it to zero. Is there a IDL way to do
> it faster than 2 for loops?
>
> thanx
>
> cheers,
> -cd

You can vectorize this.

neighborhood = [-9,-1,1,9] ;In this example, a Von Newman neighborhood

centered on the cell [0] of a 10 * x array. Do NOT keep the "central"
cell index.

data = ... ;your 2D array

neighborhoodIndices = neighborhood + indgen(n_elements(data)) ;If you
want to omit some pixels, like on the edge, put an array of valid
indices instead of the indgen().

neighborhoodValues = data[neighborhoodIndices]

sortedNeighbIndices = sort_ND(neighborhoodValues,1) ;Get sort_ND from
the web or from JD..

highestValue =
 neighborhoodValues[sortedNeighbIndices[n_elements(neighborho od)-1, *]]

writeZeroAt = where(neighborhoodValues gt data)

data[writeZeroAt] = 0

Jean
