Subject: Re: checking for connectedness of a given set of pixels
Posted by guillermo.castilla.ca on Sun, 14 Nov 2010 21:56:47 GMT
View Forum Message <> Reply to Message

On Nov 8, 3:22 pm, James <donje...@gmail.com> wrote:

> One problem is that LABEL_REGION runs over a whole array even if it is
> mostly empty (sparse). This is why LABEL_REGION might be slow for
> your task - it could be doing a lot of unnecessary extra work.

Thanks a lot James, very insightful comments, and very nice piece of code :). After giving it some more thought, I found out that there is an IDL function called REGI0N_GROW that may be faster than LABEL_REGION for this purpose when the minimum bounding box for the input set of pixels is large. I have modified your function to include the former as an alternative method (below). I'll do some tests and report back (hopefully within this year :).

```
function ISCONNECTED, points, method=method, _extra=ex
  mins = min(points, max=maxs, dimension=2)-1
  dims=maxs-mins+3
  npts= n_elements(points)/2
  indices= points - mins # replicate(1, npts)
  indices= indices[0,*] + dims[0]*indices[1,*]
  arr = bytarr(dims)
  arr[indices] = 1B
  if ~keyword_set(method) then                    $
  return, max(label_region(arr, _extra=ex)) le 1   $
  else return, n_elements(region_grow(            $
  arr, indices[0], thresh=[1B,1B], _extra=ex)) eq npts
end
```