
Subject: Re: about label_region

Posted by [Martin Downing](#) on Wed, 11 Jul 2001 11:25:25 GMT

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

Julia,

The algorithm will be written in C as it requires a raster search through the image for connected blobs. I cant find a reference right now but it goes something like this psuedo code (considering the raster order to be left to right working through rows down to up):

label_image, IMAGE

create a long image LAB_IM for output

for each pixel:

```
    ;check IMAGE to see if background pixel
    if background then continue
    ; else check LAB_IMAGE for connectivity and labelling
    if connected to labelled pixel BELOW or LEFT (also BELOW-LEFT,
    BELOW-RIGHT if 8 connected) then begin
        if connected to more than one label add label pairs to an equivalent
        list, ie [1,5] means LABEL 1 = LABEL 5
        assign (first) label to pixel in lab_image
    else assign new label to pixel in lab_image
endfor
```

use equivalent list to create a lookup table reducing all used labels to a unique set starting from 1

now run through label image replacing each label with its reduced value

hope this make some sense

Martin

--

Mr. Martin Downing,
Clinical Research Physicist,
Orthopaedic RSA Research Centre,
Woodend Hospital,
Aberdeen, AB15 6LS.

"Julia" <julia65201@usa.net> wrote in message
news:f5bebc4d.0107100654.5c89f298@posting.google.com...

>> Amar Nayegandhi said,

```
>> [  
>> label_region uses a 'connected components' algorithm that is described  
>> in any good Image Processing text book. It has the option of using 4  
>> neighbors or 8 neighbors. I have used labe_region often, and I have a  
>> strong feeling i get the same results using the 4 or 8-neighbor  
>> approach. I dont think the 8-neighbor approach gives any better results  
>> than the 4-neighbor approach.  
>> - amar  
>> ]  
>  
> Thanks for your information!  
>  
> I just wonder where IDL put all these functions in. Is it written in  
> IDL or C? I tried to realized it in IDL. As you know, since  
> connectivity algorithm is not a parallel algorithm, it should be done  
> on pixels one by one. And to keep the equivalent region information, I  
> assign a very big matrix even it is very sparse. So the algorithm is  
> very slow. Can you give me some hint to improve it, such as how to  
> keep the equivalent region information efficiently?  
>  
> Julia
```
