Subject: about label_region
Posted by julia65201 on Sun, 01 Jul 2001 22:02:14 GMT
View Forum Message <> Reply to Message

hi, there,

Yesterday, I post a message. Since it has the same subject of the
previous one. It got confused. So I bother to post it again. Wish I
can get good results.

The ones who used label_region in IDL must be suprised at the
efficieny of this function. But how can I find out what algorithm is
used in LABEL_REGION? Since I do not see this function in the library,
I assume it is written in C somewhere.

Julia

Subject: Re: about label_region
Posted by Amar Nayegandhi on Thu, 05 Jul 2001 16:22:59 GMT
View Forum Message <> Reply to Message

> The ones who used label_region in IDL must be suprised at the
> efficieny of this function. But how can I find out what algorithm is
> used in LABEL_REGION? Since I do not see this function in the library,
> I assume it is written in C somewhere.
[
 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
]

--
****
Murphy's Law of Research:
Enough research will tend to support your theory.
****

Amar Nayegandhi
Graduate Student
Department of Computer Science,
University of South Florida, Tampa.

Subject: Re: about label_region
Posted by julia65201 on Tue, 10 Jul 2001 14:54:57 GMT
View Forum Message <> Reply to Message

> 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

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