Subject: Re: connected component labeling problem in a graylevel image without background
Posted by Julia on Fri, 11 Oct 2002 16:13:33 GMT
View Forum Message <> Reply to Message

Thanks for your reply, Ben.

But I think you kind of misunderstood my problem, maybe due to my not very clear expression. :)

>> " The problem is like a grayscale photograph of a jar of
>> marbles. Each marble is uniformly gray. All the marbles are touching each
>> other, so there is no
>> background. Two marbles of the same color may not belong to the same
>> region.
>> I want to give a unique label to each marble/region."

In my problem, all the marbles are touching each other, so there is no background there. I do not think I can use
label_region on the original image. So I do like this:
First, use HISTOGRAM to get a mask of regions at each gray level, and then use LABEL_REGION on each mask.

If there are n gray levels in the image, I need do label_region n times. I think this is not very efficient. I am not sure if I can solve this kind of problem more efficiently in IDL, I call it "connected component labeling problem in a graylevel image without background".

Any suggestion there?

Julia

----- Original Message -----
From: "Ben Tupper" <btupper@bigelow.remove.org>
Newsgroups: comp.lang.idl-pvwave
Sent: Friday, October 11, 2002 10:20 AM
Subject: Re: connected component labeling problem in a graylevel image without background

> Hi there in the 'Show Me' state! My brother lives just south of
> Rolla - near Fort Leonard Wood. I have visited there once and I was
> just wowed by the geological landscape. Here in New England the
> landscape is just such dull (but beautiful, too) post-glacial!
>
> Happy to help if I can.
>

> Your problem is a bit different from that I have encountered. In my
> cases, we had a segmented image of phytoplankton and ever-present
> debris, too. Each feature (aka blob) has varying gray scale values.
> It sounds like your marbles have homogeneous gray scale values even
> though the gray scale value might change from marble to marble.
>
> I'm not sure of the specifics of what you want to do with each
> feature, but here's what I would pull them out of the image.
>
> Starting with your segmented image (background = 0, foreground = any
> values other than zero)... which has dimension nx,ny.
>
> Create a blank image that is one pixel wider in each direction than
> your segmented image. Be sure the extra pixels in the
> one-pixel-wide-pad are all set to the background value, 0. If you
> know the images will ALWAYS be the same size, simply make one copy of
> this and use it repeatedly rather than making this larger image over
> and over again. You could store this in a pointer, an object, or in a
> system variable.
>
> padded = bytarr(nx+2, ny+2)
>
> Copy your segmented image into the slightly larger image
>
> padded[1:nx-2, 1:ny-2] = segmented
>
> Run the padded image through LABEL_REGION
>
> labeled = LABEL_REGION(padded, keywords=keywords)
>
> Use HISTOGRAM to get the indices of the color blobs - subset the
> labeled image at the same time. Capture the reverse indices.
>
> h = HISTOGRAM(labeled[1:nx, 1:ny], reverse_indices = r)
>
> Now through the magic of reverse_indices, you can quickly pull out the
> location of each feature (marble, in this case). I can't recall the
> syntax off hand (I don't have IDL right here, either), but I do recall
> an example in the documentation for HISTOGRAM. If that doesn't help,
> check out David Fanning's web pages.
>
> I'm not sure if that helps or answers your question. I'm not even
> sure, now that I reread you note, what your question is. This does
> mean that you have scanned the image at least twice (LABEL_REGION and
> HISTOGRAM) in addition to any scanning you did to segment the image.
>
> Cheers,
> Ben

>>