

---

Subject: perimeter of a blob

Posted by [Ben Tupper](#) on Wed, 26 Jan 2000 08:00:00 GMT

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

---

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
```

```
<html>
```

I am looking for an efficient means of finding the perimeter of contiguous blobs of pixels within an image. These blobs are the regions returned by the LABEL\_REGION function. I have been using the trusty brute-force-and-ignorance approach where I scan rows and columns within each blob, recording the min and max coordinates of each row/column. It is a cumbersome task and leaves gaps in certain areas (especially concavities in the blob.)

<p>I'd like to try something that does the following:

<p>Input image and blob info -> Something mysterious happens inside the computer -> Output the index values of the perimeter of the blob

<p>I think this closely approximates the reverse of POLYFILLV (pretty close anyhow.)

<p>Any suggestions?

<p>Thanks in advance,

<p>Ben

```
<pre>--&nbsp;
```

Ben Tupper

Bigelow Laboratory for Ocean Science

tupper@seadas.bigelow.org

```
pemaquidriver@tidewater.net</pre>
```

```
&nbsp;</html>
```

---

---

Subject: Re: perimeter of a blob

Posted by [Ben Tupper](#) on Thu, 27 Jan 2000 08:00:00 GMT

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

---

Thanks to each of you for your help.

I have tried the convolution and contouring methods suggested by Richard and David respectively. I used a black (0) and white (255) image of the capital letter G (hand drawn.)

The convolution exactly defines the inclusive perimeter of the blob. I haven't figured out how to sort the indices into a clockwise (or counter clockwise order yet... but I haven't tried either.)

The contour method misses the exact inclusive perimeter whenever an inside corner is encountered (i.e. concavity.) It cuts diagonally

across these inside corners a result. The loss is nearly insignificant in this case, but I recall have difficulty comparing polygon perimeter estimates made in IDL with those made on the same data in ARC/INFO.

Each method works for my current purposes.

Thanks again!

Ben

--

Ben Tupper  
Pemaquid River Company  
248 Lower Round Pond Road  
POB 106  
Bristol, ME 04539

Tel: (207) 563-1048  
Email: [PemaquidRiver@tidewater.net](mailto:PemaquidRiver@tidewater.net)

---

Subject: Re: perimeter of a blob  
Posted by [davidf](#) on Thu, 27 Jan 2000 08:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Richard Adams ([r.j.adams@bath.ac.uk](mailto:r.j.adams@bath.ac.uk)) writes:

> Dear Ben, if you just need the indices and not the indices in order around  
> the perimeter then you could use an imaging method like this:

You know, I think this whole thing has to be a lot easier than this.

Ben is getting the data from Label\_Region, which returns regions labeled with a single value (or "color", as I prefer to think of them). Why not just use the CONTOUR command to draw a contour around the region, then use the information in all those PATH\_\*\*\*\* keywords to find the perimeter? With clever programming (I'll leave this part to Ben) you can find all the holes in the data, etc.

I think I might even have a piece of code around here where I did this once. If I get some time (unlikely today) I'll have a look around.

Cheers,

David

--

David Fanning, Ph.D.  
Fanning Software Consulting  
Phone: 970-221-0438 E-Mail: davidf@dfanning.com  
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>  
Toll-Free IDL Book Orders: 1-888-461-0155

---

---

Subject: Re: perimeter of a blob  
Posted by [Richard Adams](#) on Thu, 27 Jan 2000 08:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear Ben, if you just need the indices and not the indices in order around the perimeter then you could use an imaging method like this:

```
a = your image; assume it is one blob and has value 255 as object
otherwise 0
bb = convol(a, replicate(1,3,3), 9, /center); change the kernel if you
want some other connectivity
edges = where(bb gt 0 and bb lt 255); these are edges of your blob both
inside and out

bb[*] = 0
bb[edges] = 255 ; make this new image just edges
bb = A AND BB ; now keep just those edges inside your object
perimeter = where(bb eq 255); these are indices to final outline
```

I think this works - does at least on dumb test images. If you want them in order then I guess you could trace around with a maze-search algorithm. Big disadvantage is that you need to isolate a blob at a time, I think. Interested to hear if it works around all your corners.

Richard.

--

Richard J Adams           }<}}}}i¿½>           e: r.j.adams@bath.ac.uk  
MRC Senior Research Fellow           t: +44 1225 826436  
Department of Biology and Biochemistry   f: +44 1225 826779  
University of Bath, Bath, BA2 7AY, UK

---