Subject: Re: pixel coordinates of a line
Posted by marc schellens[1] on Thu, 28 Jun 2001 06:35:02 GMT
View Forum Message <> Reply to Message

Oh, I forgot:
I have the start and endpoint, but they might need to be clipped
to the images boders...

> I want to extract pixels along a line.
> I know start and endpoint (ie. I draw the line in top of the
> image).
> How to get the pixel values? Do I have to do it 'by hand'?
>
> thanks,
> marc

Subject: Re: pixel coordinates of a line
Posted by Alex Schuster on Thu, 28 Jun 2001 12:43:09 GMT
View Forum Message <> Reply to Message

Marc Schellens wrote:

> I want to extract pixels along a line.
> I know start and endpoint (ie. I draw the line in top of the
> image).
> How to get the pixel values? Do I have to do it 'by hand'?

Good question. I don't know about such a function in IDL, but I think
there should be a routine for that.

What about this:
Your line runs from (x1,y1) to (x2,y2), the image has a size of
dimx*dimy. Let's create a thin, one pixel wide polygon along this line:
index = polyfillv([x1,x2,x2+1,x1+1], [y1,y2,y2,y1], dimx, dimy)
(This assumes that |x2-x1| lt |y2-y1|).
Your pixel values are image[index] then. Well, nearly.

Another method would be drawing a real line via PLOTS (preferably in the
Z buffer), use TVRD() to get this screen part, use WHERE() to get the
indices of this line, and again image[index] is the information you
seek.

But let's wait a little for other responses, I'm pretty sure J.D. will
come up with another idea involving HISTOGRAM().

Alex
--

Alex Schuster Wonko@planet-interkom.de
alex@pet.mpin-koeln.mpg.de

Subject: Re: pixel coordinates of a line
Posted by tam on Thu, 28 Jun 2001 13:33:54 GMT
View Forum Message <> Reply to Message

Marc Schellens wrote:
>
> Oh, I forgot:
> I have the start and endpoint, but they might need to be clipped
> to the images boders...
>
>> I want to extract pixels along a line.
>> I know start and endpoint (ie. I draw the line in top of the
>> image).
>> How to get the pixel values? Do I have to do it 'by hand'?
>>
>> thanks,
>> marc

Working in pixel coordinates you have the endpoints (sx,sy,ex,ey) and the clipping rectangle (x0,y0, x1, y1), so you should trivially be able to get the equation of the line

```
y = m x + b    [ m= (ey-sy)/(ex-sx), b = sy - m sx ]
```

Turn this into IDL...

```
x = lindgen(x1-x0) + x0
y = floor (m * x + b + 0.5)    [add 0.5 to get appropriate rounding]
w = where (y ge y0 and y le y1)
if (w[0] ne -1) then begin
   x = x(w)
   y = y(w)
endif else begin
   .... line is outside clipping rectangle
endelse
```

Here I'm assuming integral values for x0,x1, y0,y1 but not necessarily the end points.

There are probably subtleties I've missed but this seems like a reasonable start. Whether it will exactly match the pixels that IDL will draw on is hard to say... Guess we'd need to know exactly how IDL draws a line.

Tom McGlynn
NASA/GSFC
tam@lheapop.gsfc.nasa.gov
[Mailed and posted]

Subject: Re: pixel coordinates of a line
Posted by david[2] on Thu, 28 Jun 2001 14:42:34 GMT
View Forum Message <> Reply to Message

Marc Schellens writes:

> I want to extract pixels along a line.
> I know start and endpoint (ie. I draw the line in top of the
> image).
> How to get the pixel values? Do I have to do it 'by hand'?

Here is an article that describes how to do this
for another purpose, using the INTERPOLATE function:

http://www.dfanning.com/tips/image_profile.html

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: pixel coordinates of a line
Posted by Craig Markwardt on Thu, 28 Jun 2001 15:10:39 GMT
View Forum Message <> Reply to Message

david@dfanning.com (David Fanning) writes:
>
> Here is an article that describes how to do this
> for another purpose, using the INTERPOLATE function:
>
> http://www.dfanning.com/tips/image_profile.html

I use the INTERPOLATE method quite a bit. It has the advantage that

you get the benefit of immediately nearby points, and you get a smooth
curve with as much resolution as you desire.

Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------

Subject: Re: pixel coordinates of a line
Posted by marc schellens[1] on Fri, 29 Jun 2001 08:45:15 GMT
View Forum Message <> Reply to Message

Thanks to everybody.

Tom's method of course would work,
but I don't like to do such things in IDL 'by hand'.
Even in my case now, speed is not a matter.

Alex method with the polygon was really interesting.
Even I am not sure about the clipping here.
But I ended up using the z buffer method.

The interpolate routine is also a good suggestion,
but also doesn't consider the clipping.

Anyway,
some useful tips for further problems,
thanks again,
:-) marc

Subject: Re: pixel coordinates of a line
Posted by Craig Markwardt on Fri, 29 Jun 2001 13:22:22 GMT
View Forum Message <> Reply to Message

Marc Schellens <m_schellens@hotmail.com> writes:

> The interpolate routine is also a good suggestion,
> but also doesn't consider the clipping.

INTERPOLATE does consider clipping if you use the MISSING keyword.

Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------

Subject: Re: pixel coordinates of a line
Posted by david[2] on Fri, 29 Jun 2001 13:53:14 GMT
View Forum Message <> Reply to Message

Alex Schuster writes:

> Another method would be drawing a real line via PLOTS (preferably in the
> Z buffer), use TVRD() to get this screen part, use WHERE() to get the
> indices of this line, and again image[index] is the information you
> seek.

Oddly, I've had occasion to use this method in the past
week or so and I want to mention one little caveat.
I was trying to quantitate the length of a "contact"
between two adjoining "blobs". The "contact" shows
up (more or less) as a line on an image.

But since I wanted to measure the length of this
line, I needed the pixels in order, from one end
to the other. WHERE doesn't give you this. Nor
does the very useful routine THIN, which has the
wonderful property of identifying the endpoints
of the line for you. (Nor does SEARCH2D return
ordered pixels, which was another non-productive
thought I had.)

So, if you want a line segment, you will have
to write a little routine to sort the pixels
out for yourself. My routine starts at one of
the endpoints and finds the next closest pixel,
etc. It may not be the fastest algorithm, but
it worked well for me.

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