Subject: Regarding the Fit_Ellipse Program
Posted by plim.dreaming on Tue, 26 Feb 2008 14:28:22 GMT
View Forum Message <> Reply to Message

Greetings all!
The last time I posted here I got awesome help and I have what I believe is a smaller problem this time.
I am trying to use the fit_ellipse program (http://www.dfanning.com/ip_tips/fit_ellipse.html) and having some difficulty in that my ROI is a contour defined by a 2-D array and not a 1-D array of pixels.
My 2-D array contains the x and y positions of all the points along that contour (or ROI) so I am trying to translate this fit_ellipse program to something I can use.

Would really appreciate the help!
Bruno

Subject: Re: Regarding the Fit_Ellipse Program
Posted by pgrigis on Tue, 26 Feb 2008 21:51:51 GMT
View Forum Message <> Reply to Message

plim.dream...@gmail.com wrote:
> So, I used contour to obtain the x and y positions of a certain
> contour level and labeled those arrays as posx and posy.
> then obtained the positions in terms of the device, because that is
> what polyfillv uses:
> D = convert_coord(posx,posy, /to_device)
>
>
> I did a print,!D.x_size to get the device sizes (it gave me the famous
> 17780) so I did:

This happens most likely because you are using the postscript device.
If you do set_plot,'X' or 'WIN', you should get more resonable values for !D.x_size.

Ciao,
Paolo

> result = polyfillv(D(0,*),D(1,*),17780,17780)
> plots,fit_ellipse(result)
>
> It is unable to allocate memory for the fit_ellipse part.
> Is the !D.x_size related to the monitor size or the monitor resolution
> or the postscript resolution or non-of-the-above?
>
> And with the IDLanROI I finally made the object

> blob = OBJ_NEW('IDLanROI', posx, posy) and have to say that I
> certainly was/am scared by wording such as: "Init methods are special
> life cycle methods..."!?
> I am right now trying to figure out how to type in this new language
> Result = Obj->[IDLanROI::]ComputeMask(
> to get from here to fitting an ellipse to the object and getting the
> semi-major and semi-minor axis.
>
> Much thanks,
> "Bruno Schmidt of Northern Florida State University"
>
> - so honey what did you do at work all day today?
> - oh, i drew an ellipse...

Subject: Re: Regarding the Fit_Ellipse Program
Posted by pgrigis on Tue, 26 Feb 2008 22:06:07 GMT
View Forum Message <> Reply to Message

plim.dream...@gmail.com wrote:
> So, I used contour to obtain the x and y positions of a certain
> contour level and labeled those arrays as posx and posy.
> then obtained the positions in terms of the device, because that is
> what polyfillv uses:
> D = convert_coord(posx,posy, /to_device)
>
> I did a print,!D.x_size to get the device sizes (it gave me the famous
> 17780) so I did:
> result = polyfillv(D(0,*),D(1,*),17780,17780)
I think you should do this operation in data or normalized space
instead,
and just chose a suitable array dimension.

Ciao,
Paolo

> plots,fit_ellipse(result)
>
> It is unable to allocate memory for the fit_ellipse part.
> Is the !D.x_size related to the monitor size or the monitor resolution
> or the postscript resolution or non-of-the-above?
>
> And with the IDLanROI I finally made the object
> blob = OBJ_NEW('IDLanROI', posx, posy) and have to say that I
> certainly was/am scared by wording such as: "Init methods are special
> life cycle methods..."!?
> I am right now trying to figure out how to type in this new language
> Result = Obj->[IDLanROI::]ComputeMask(

> to get from here to fitting an ellipse to the object and getting the
> semi-major and semi-minor axis.
>
> Much thanks,
> "Bruno Schmidt of Northern Florida State University"
>
> - so honey what did you do at work all day today?
> - oh, i drew an ellipse...

Subject: Re: Regarding the Fit_Ellipse Program
Posted by plim.dreaming on Tue, 26 Feb 2008 22:19:04 GMT
View Forum Message <> Reply to Message

On Feb 26, 4:51 pm, pgri...@gmail.com wrote:
> plim.dream...@gmail.com wrote:
>> So, I used contour to obtain the x and y positions of a certain
>> contour level and labeled those arrays as posx and posy.
>> then obtained the positions in terms of the device, because that is
>> what polyfillv uses:
>> D = convert_coord(posx,posy, /to_device)
>
>> I did a print,!D.x_size to get the device sizes (it gave me the famous
>> 17780) so I did:
>
> This happens most likely because you are using the postscript device.
> If you do set_plot,'X' or 'WIN', you should get more resonable values
> for !D.x_size.
>
> Ciao,
> Paolo> result = polyfillv(D(0,*),D(1,*),17780,17780)
>> plots,fit_ellipse(result)

Hi Paolo, Thanks for the reply,
you are right, and yet I do want to put the ellipse onto the
postscript output.
B

Subject: Re: Regarding the Fit_Ellipse Program
Posted by pgrigis on Tue, 26 Feb 2008 22:28:49 GMT
View Forum Message <> Reply to Message

plim.dream...@gmail.com wrote:
> On Feb 26, 4:51 pm, pgri...@gmail.com wrote:
>> plim.dream...@gmail.com wrote:
>>> So, I used contour to obtain the x and y positions of a certain

>>> contour level and labeled those arrays as posx and posy.
>>> then obtained the positions in terms of the device, because that is
>>> what polyfillv uses:
>>> D = convert_coord(posx,posy, /to_device)
>>
>>> I did a print,!D.x_size to get the device sizes (it gave me the famous
>>> 17780) so I did:
>>
>> This happens most likely because you are using the postscript device.
>> If you do set_plot,'X' or 'WIN', you should get more resonable values
>> for !D.x_size.
>>
>> Ciao,
>> Paolo> result = polyfillv(D(0,*),D(1,*),17780,17780)
>>> plots,fit_ellipse(result)
>
> Hi Paolo, Thanks for the reply,
> you are right, and yet I do want to put the ellipse onto the
> postscript output.
Yes, so what you need to do is to compute the ellipse
in a way that does not depend on the graphic device used and
then plot it. Can I just quickly ask what is your starting data
again (that is, how exactly you define your points in the first
place)?

Paolo

> B

Subject: Re: Regarding the Fit_Ellipse Program
Posted by David Fanning on Tue, 26 Feb 2008 23:16:48 GMT
View Forum Message <> Reply to Message

pgrigis@gmail.com writes:

> I think you should do this operation in data or normalized space
> instead,
> and just chose a suitable array dimension.

Now, there's a good idea. I was just telling my class
last week about the large number of people I know who
use GhostView as their standard monitor, and they just
stared at me, dumbfounded.

But, yes, using device coordinates in PostScript is
probably not what I would recommend. Your arrays could
get, uh, large. :-)

"Life-cycle method" is a fancy way of saying the INIT
and CLEANUP methods are not called directly. They are
called indirectly when you create (OBJ_NEW) and destroy
(OBJ_DESTROY) your object.

Maybe you could show us a bit of code, so we could
suggest something reasonable.

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Regarding the Fit_Ellipse Program
Posted by plim.dreaming on Wed, 27 Feb 2008 23:47:29 GMT
View Forum Message <> Reply to Message

> Yes, so what you need to do is to compute the ellipse
> in a way that does not depend on the graphic device used and
> then plot it. Can I just quickly ask what is your starting data
> again (that is, how exactly you define your points in the first
> place)?
>
> Paolo> B

Hi Paolo and David,
So, I went ahead an managed to create ellipses for the blobs!! I
changed set_plot,'x' so now !D.X_size gives me 640 instead which the
computer can crunch and produce nice figures on the screen. I would
still like to convert the values for x,y, semimajor, semiminor, and
center of the ellipse back to data coordinates such that I can place
them on the postscript file again but convert_coord(... /to_data)
doesn't do the job.

Some code to make this more interesting:

contour_array = [s2c2(0,*),s2c2(1,*)] ; this is the x,y positions of
the vertices of the polygon which defines the contour

blob_x = s2c2(0,*)
blob_y = s2c2(1,*)

```
set_plot,'x'

polyfill,blob_x,blob_y,color=100,/data

D = convert_coord(blob_x, blob_y, /to_device)

result = polyfillv(d(0,*), d(1,*),640,640)

Ellipse_out = fit_ellipse(result,semiaxes=semiaxes,center=center)

plots,Ellipse_out,/device

  set_plot,'ps
  device,filename='Surface_Plots.ps'
New_coords = convert_coord(Ellipse_out, /to_device)
     plots,s2c2(0,*),s2c2(1,*)
plots,New_coords,/device
  device,/close
```

Much thanks!
Bruno

p.s. I haven't banged my head on the IDLanROI more than being capable
of making the object:
blob_roi = OBJ_NEW('IDLanROI', blob_x,blob_y)
blob_mask = blob_roi->ComputeMask(initialize=0)
because I still don't see how to fit an ellipse and obtain its
parameters using the object functions.

Subject: Re: Regarding the Fit_Ellipse Program
Posted by pgrigis on Wed, 27 Feb 2008 23:56:13 GMT

View Forum Message <> Reply to Message

plim.dream...@gmail.com wrote:
>> Yes, so what you need to do is to compute the ellipse
>> in a way that does not depend on the graphic device used and
>> then plot it. Can I just quickly ask what is your starting data
>> again (that is, how exactly you define your points in the first
>> place)?
>>
>> Paolo> B
>
>
> Hi Paolo and David,
> So, I went ahead an managed to create ellipses for the blobs!! I
> changed set_plot,'x' so now !D.X_size gives me 640 instead which the

> computer can crunch and produce nice figures on the screen. I would
> still like to convert the values for x,y, semimajor, semiminor, and
> center of the ellipse back to data coordinates such that I can place
> them on the postscript file again but convert_coord(... /to_data)
> doesn't do the job.
>
> Some code to make this more interesting:
>
> contour_array = [s2c2(0,*),s2c2(1,*)] ; this is the x,y positions of
> the vertices of the polygon which defines the contour

May ask again what are your staring points, before you do the contour?
How are these points defind? I have this feeling that there may be a
much
simpler way...

Paolo

>
> blob_x = s2c2(0,*)
> blob_y = s2c2(1,*)
>
> set_plot,'x'
>
> polyfill,blob_x,blob_y,color=100,/data
>
> D = convert_coord(blob_x, blob_y, /to_device)
>
> result = polyfillv(d(0,*), d(1,*),640,640)
>
> Ellipse_out = fit_ellipse(result,semiaxes=semiaxes,center=center)
>
> plots,Ellipse_out,/device
>
> set_plot,'ps
> device,filename='Surface_Plots.ps'
> New_coords = convert_coord(Ellipse_out, /to_device)
> plots,s2c2(0,*),s2c2(1,*)
> plots,New_coords,/device
> device,/close
>
> Much thanks!
> Bruno
>
> p.s. I haven't banged my head on the IDLanROI more than being capable
> of making the object:
> blob_roi = OBJ_NEW('IDLanROI', blob_x,blob_y)
> blob_mask = blob_roi->ComputeMask(initialize=0)

> because I still don't see how to fit an ellipse and obtain its
> parameters using the object functions.

Subject: Re: Regarding the Fit_Ellipse Program
Posted by David Fanning on Thu, 28 Feb 2008 00:00:20 GMT
View Forum Message <> Reply to Message

pgrigis@gmail.com writes:

> May ask again what are your staring points, before you do the contour?
> How are these points defind? I have this feeling that there may be a
> much
> simpler way...

I agree. You shouldn't have to work so hard to
get PostScript output. It should be as simple
as setting plot to "PS" and running your program
again.

But Paolo and I are working blind. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming (www.dfanning.com)
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Regarding the Fit_Ellipse Program
Posted by plim.dreaming on Thu, 28 Feb 2008 13:46:39 GMT
View Forum Message <> Reply to Message

> May ask again what are your staring points, before you do the contour?
> How are these points defind? I have this feeling that there may be a
> much
> simpler way...
>
> Paolo

Hi Paolo and David,
Hummm, well, before I do the contour I just have a square field full
of points, so a big array of x,y (this is input data, I do not choose
the locations of these points, i read them in from a file), and to

each point I have a density-value attributed, rho. The points are not
in a matrix, they are randomly distributed in the field, so I do a:
contour,rho,x,y,levels=level,triangulation=tri,path_xy=polic ons2,path_info=infocons2,/
path_data_coords,/path_doublt,/noerase,/data
That provides me with the xy positions of the vertices of several
different contours in the field all of which have the same density
level. For each contour I want to fit and plot an ellipse and obtain
the parameters.
Hope this helps,
Bruno

Subject: Re: Regarding the Fit_Ellipse Program
Posted by pgrigis on Thu, 28 Feb 2008 15:44:13 GMT
View Forum Message <> Reply to Message

plim.dream...@gmail.com wrote:
>> May ask again what are your staring points, before you do the contour?
>> How are these points defind? I have this feeling that there may be a
>> much
>> simpler way...
>>
>> Paolo
>
> Hi Paolo and David,
> Hummm, well, before I do the contour I just have a square field full
> of points, so a big array of x,y (this is input data, I do not choose
> the locations of these points, i read them in from a file), and to
> each point I have a density-value attributed, rho. The points are not
> in a matrix, they are randomly distributed in the field, so I do a:
> contour,rho,x,y,levels=level,triangulation=tri,path_xy=polic ons2,path_info=infocons2,/
> path_data_coords,/path_doublt,/noerase,/data

I don't think this is the right approach to the problem.
What I would do is to get a regularly gridded array
for rho first (for instance with trigrid, but there may
be other possibilities).
Once you have that, it is easy to produce an input array
suitable for fit_ellipse (just set all points higher (or lower)
than a threshold to 0, and the others to 1).

Try something like:

TRIANGULATE, x, y, Triangles
Result = TRIGRID(x,y,rho, Triangles,[dx,dy],[minx,miny,maxx,maxy]])

where dx and dy is the spacing of the regular grid and minx etc. are
the boundaries.

Cheers,
Paolo

> That provides me with the xy positions of the vertices of several
> different contours in the field all of which have the same density
> level. For each contour I want to fit and plot an ellipse and obtain
> the parameters.
> Hope this helps,
> Bruno

Subject: Re: Regarding the Fit_Ellipse Program
Posted by plim.dreaming on Thu, 28 Feb 2008 16:30:04 GMT
View Forum Message <> Reply to Message

> I don't think this is the right approach to the problem.
> What I would do is to get a regularly gridded array
> for rho first (for instance with trigrid, but there may
> be other possibilities).
> Once you have that, it is easy to produce an input array
> suitable for fit_ellipse (just set all points higher (or lower)
> than a threshold to 0, and the others to 1).
>
> Try something like:
>
> TRIANGULATE, x, y, Triangles
> Result = TRIGRID(x,y,rho, Triangles,[dx,dy],[minx,miny,maxx,maxy]])
>
> where dx and dy is the spacing of the regular grid and minx etc. are
> the boundaries.
>
> Cheers,
> Paolo>

Hi Paolo,
I understand that it would be simpler to use a regular gridded array
but that is changing the whole situation.
Part of the study I am doing is based on the fact that it isn't a
regular gridded array with defined spacings dx and dy.
It seems we are straying from the problem which would be to fit an
ellipse to a contour and obtain its parameters given the x and y
positions of the contour. I'm getting amazed at how hard it is to fit
an ellipse to it!
I certainly don't understand why there isn't a fit_ellipse program
that works with the x,y array of the contour vertices instead of the
pixel indices of the image of a contour (I wish I knew how to make one
myself).

B

Subject: Re: Regarding the Fit_Ellipse Program
Posted by pgrigis on Thu, 28 Feb 2008 16:53:34 GMT
View Forum Message <> Reply to Message

plim.dream...@gmail.com wrote:
>> I don't think this is the right approach to the problem.
>> What I would do is to get a regularly gridded array
>> for rho first (for instance with trigrid, but there may
>> be other possibilities).
>> Once you have that, it is easy to produce an input array
>> suitable for fit_ellipse (just set all points higher (or lower)
>> than a threshold to 0, and the others to 1).
>>
>> Try something like:
>>
>> TRIANGULATE, x, y, Triangles
>> Result = TRIGRID(x,y,rho, Triangles,[dx,dy],[minx,miny,maxx,maxy]])
>>
>> where dx and dy is the spacing of the regular grid and minx etc. are
>> the boundaries.
>>
>> Cheers,
>> Paolo>
>
> Hi Paolo,
> I understand that it would be simpler to use a regular gridded array
> but that is changing the whole situation.
> Part of the study I am doing is based on the fact that it isn't a
> regular gridded array with defined spacings dx and dy.

Exactly, so I was suggesting you *transform* your irregularly gridded
data into a regular grid (which is probably what contour is doing
behind
the curtains anyway, only is a bit harder to know and control what
contour is doing)!
Try this code:

```
np=1000
;creates irregular grid
x=randomu(seed,np)*4-2
y=randomu(seed,np)*4-2

;creates some mock data
a=2
b=3
```

```
z=1-sqrt((x/a)^2+(y/b)^2)
z=z+randomn(seed,np)*z*0.1

;this is the data the way contour sees it
contour,z,x,y,/irregular,/iso,levels=findgen(11)/10.*1.15
oplot,x,y,psym=6

;put the data into a regular grid
TRIANGULATE, X, Y, Triangles
Result = TRIGRID( X, Y, Z, Triangles,[0.01,0.01],[-3.,-3,3,3] )

;show regularly gridded data
loadct,5
tvscl,result

;fit ellipse to points above 0.5
ind=where(result GT 0.5)
ell=result*0
ell[ind]=1
data=fit_ellipse(ind,xsize=602,ysize=602)

;show data & best-fit ellipse
tvscl,ell
plots,data[0,*],data[1,*],color=128,/device,thick=2
```

I think this pretty much does what you want, right?

Paolo

> It seems we are straying from the problem which would be to fit an
> ellipse to a contour and obtain its parameters given the x and y
> positions of the contour. I'm getting amazed at how hard it is to fit
> an ellipse to it!
> I certainly don't understand why there isn't a fit_ellipse program
> that works with the x,y array of the contour vertices instead of the
> pixel indices of the image of a contour (I wish I knew how to make one
> myself).
> B

Subject: Re: Regarding the Fit_Ellipse Program
Posted by pgrigis on Thu, 28 Feb 2008 17:10:25 GMT
View Forum Message <> Reply to Message

plim.dream...@gmail.com wrote:

[skip]

> I certainly don't understand why there isn't a fit_ellipse program
> that works with the x,y array of the contour vertices instead of the
> pixel indices of the image of a contour (I wish I knew how to make one
> myself).

In general, if you have a number of x and y coordinates for the
boundary
of the ellipse it is not possible to directly use the method used
in fit_ellipse, beacuse if there are, say, twice as many points in
the left side of the ellipse than in the right side, the center of
mass
and moment of inertia will be shifted to the left.

One could then try to "fill out" a two dimensional shape.
Using polyfillv is not a good idea because the results depend in
general on the ordering of the x and y vector (if I swap two
points, the polygon will be different, but the best fit ellipse should
not depend on ordering or labeling of the points).
One could try and use a convex hull insetad, but this may lead to
overestimation of the size of the ellipse and would be really badly
sensitive to outliers.

In such a situation (given unsorted x,y coordinates)
I think that it would probably better to use a minimization function,
with the weight being minimized as the sum of the squares of the
distances of each point to the ellipse. This should not be too hard
to implement.

Paolo

> B

Subject: Re: Regarding the Fit_Ellipse Program
Posted by David Fanning on Thu, 28 Feb 2008 17:27:00 GMT
View Forum Message <> Reply to Message

plim.dreaming@gmail.com writes:

> I certainly don't understand why there isn't a fit_ellipse program
> that works with the x,y array of the contour vertices instead of the
> pixel indices of the image of a contour (I wish I knew how to make one
> myself).

The problem with using the contour vertices, is that

they are not evenly spaced, so the "weighting" of
the ellipse would be wildly off. I do have a program,
named ARCSAMPLE, that can resample "reasonable" curves,
so that you have approximately equally spaced points
along the curve. Depending on what your "blob" looks
like, you might prefer to use this for calculating the
ellipse.

http://www.dfanning.com/programs/arcsample.pro

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming (www.dfanning.com)
Sepore ma de ni thui. ("Perhaps thou speakest truth.")
