
Subject: ellipse fitting?

Posted by [tom](#) on Sat, 27 Apr 2002 13:52:48 GMT

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

I found a matlab function for ellipse, but it is not easy for me translate to IDL. For example,

```
% Solve eigensystem  
[gevec, geval] = eig(S,C);
```

are there any function like eig(S,C) in IDL?

The matlab for ellips fitting is as following, who have a idl version?

```
-----  
-----  
  
function a = fitellipse(X,Y)  
  
% FITELLIPSE Least-squares fit of ellipse to 2D points.  
% A = FITELLIPSE(X,Y) returns the parameters of the best-fit  
% ellipse to 2D points (X,Y).  
% The returned vector A contains the center, radii, and orientation  
% of the ellipse, stored as (Cx, Cy, Rx, Ry, theta_radians)  
%  
% Authors: Andrew Fitzgibbon, Maurizio Pilu, Bob Fisher  
% Reference: "Direct Least Squares Fitting of Ellipses", IEEE T-PAMI, 1999  
%  
% This is a more bulletproof version than that in the paper, incorporating  
% scaling to reduce roundoff error, correction of behaviour when the input  
% data are on a perfect hyperbola, and returns the geometric parameters  
% of the ellipse, rather than the coefficients of the quadratic form.  
%  
% Example: Run fitellipse without any arguments to get a demo  
if nargin == 0  
    % Create an ellipse  
    t = linspace(0,2);  
  
    Rx = 300  
    Ry = 200  
    Cx = 250  
    Cy = 150  
    Rotation = .4 % Radians  
  
    x = Rx * cos(t);  
    y = Ry * sin(t);
```

```

nx = x*cos(Rotation)-y*sin(Rotation) + Cx;
ny = x*sin(Rotation)+y*cos(Rotation) + Cy;
% Draw it
plot(nx,ny,'o');
% Fit it
fitellipse(nx,ny)
% Note it returns (Rotation - pi/2) and swapped radii, this is fine.
return
end

```

```

% normalize data
mx = mean(X);
my = mean(Y);
sx = (max(X)-min(X))/2;
sy = (max(Y)-min(Y))/2;

```

```

x = (X-mx)/sx;
y = (Y-my)/sy;

```

```

% Force to column vectors
x = x(:);
y = y(:);

```

```

% Build design matrix
D = [ x.*x x.*y y.*y x y ones(size(x)) ];

```

```

% Build scatter matrix
S = D'*D;

```

```

% Build 6x6 constraint matrix
C(6,6) = 0; C(1,3) = -2; C(2,2) = 1; C(3,1) = -2;

```

```

% Solve eigensystem
[gevec, geval] = eig(S,C);

```

```

% Find the negative eigenvalue
I = find(real(diag(geval)) < 1e-8 & ~isinf(diag(geval)));

```

```

% Extract eigenvector corresponding to negative eigenvalue
A = real(gevec(:,I));

```

```

% unnormalize
par = [
    A(1)*sy*sy, ...
    A(2)*sx*sy, ...
    A(3)*sx*sx, ...
    -2*A(1)*sy*sy*mx - A(2)*sx*sy*my + A(4)*sx*sy*sy, ...
    -A(2)*sx*sy*mx - 2*A(3)*sx*sx*my + A(5)*sx*sx*sy, ...

```

```

A(1)*sy*sy*mx*mx + A(2)*sx*sy*mx*my + A(3)*sx*sx*my*my ...
- A(4)*sx*sy*sy*mx - A(5)*sx*sx*sy*my ...
+ A(6)*sx*sx*sy*sy ...
]';

```

```

% Convert to geometric radii, and centers

```

```

thetarad = 0.5*atan2(par(2),par(1) - par(3));
cost = cos(thetarad);
sint = sin(thetarad);
sin_squared = sint.*sint;
cos_squared = cost.*cost;
cos_sin = sint .* cost;

```

```

Ao = par(6);
Au = par(4) .* cost + par(5) .* sint;
Av = - par(4) .* sint + par(5) .* cost;
Auu = par(1) .* cos_squared + par(3) .* sin_squared + par(2) .* cos_sin;
Avv = par(1) .* sin_squared + par(3) .* cos_squared - par(2) .* cos_sin;

```

```

% ROTATED = [Ao Au Av Auu Avv]

```

```

tuCentre = - Au./(2.*Auu);
tvCentre = - Av./(2.*Avv);
wCentre = Ao - Auu.*tuCentre.*tuCentre - Avv.*tvCentre.*tvCentre;

```

```

uCentre = tuCentre .* cost - tvCentre .* sint;
vCentre = tuCentre .* sint + tvCentre .* cost;

```

```

Ru = -wCentre./Auu;
Rv = -wCentre./Avv;

```

```

Ru = sqrt(abs(Ru)).*sign(Ru);
Rv = sqrt(abs(Rv)).*sign(Rv);

```

```

a = [uCentre, vCentre, Ru, Rv, thetarad];

```

Subject: Re: ellipse fitting?

Posted by [David Fanning](#) on Sun, 28 Apr 2002 00:58:06 GMT

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

tom (tom2959@21cn.com) writes:

```

> I found a matlab function for ellipse, but it is not easy for me translate
> to IDL. For example,
>
> % Solve eigensystem

```

> [gevec, geval] = eig(S,C);
>
> are there any function like eig(S,C) in IDL?
>
> The matlab for ellips fitting is as following, who have a idl version?

I've put a new IDL program named FIT_ELLIPSE on my web page:

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

In true IDL programmer fashion, I've used the ideas of others to create something useful to me. In this case, I'm particularly grateful to Craig Markwardt, who wrote the eigenvalue part of the code as a favor to me, and to Wayne Landsman, whose program TVEllipse I've used for a long time. You can find this program on the NASA Goddard IDL web page.

The Fit_Ellipse function accepts a 1D array of pixel indices, and returns the points that describe the fitted ellipse in device coordinates. The format of the points is such that they can be sent directly to PLOTS. Optional keywords allow you to obtain the center of the ellipse, the orientation of the major axis, and the major/minor or the semi-major/semi-minor axes.

Cheers,

David

--

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

Subject: Re: ellipse fitting?
Posted by [Craig Markwardt](#) on Sun, 28 Apr 2002 14:51:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

"tom" <tom2959@21cn.com> writes:

> I found a matlab function for ellipse, but it is not easy for me translate
> to IDL. For example,

>
> % Solve eigensystem
> [gevec, geval] = eig(S,C);
>
> are there any function like eig(S,C) in IDL?
>
> The matlab for ellips fitting is as following, who have a idl version?

If you have the set of (x,y) points which define the outline of the ellipse, then how about MPFITELLIPSE? There are a few auxiliary routines to download as well.

Craig

<http://cow.physics.wisc.edu/~craigm/idl/idl.html> (under Fitting)

--

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

Subject: Re: ellipse fitting?

Posted by [hradilv.nospam](#) on Mon, 29 Apr 2002 16:19:10 GMT

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

I did the same thing that you are doing a couple of months ago!

The best alternatives are

1- use mpfitellipse or Dr. Fanning's code

2- call lapack routines need to emulate the eig() function. Namely, I used the function dggev.f :

<http://www.netlib.org/lapack/double/dggev.f>

* DGGEV computes for a pair of N-by-N real nonsymmetric matrices (A,B)

* the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors.

*

* A generalized eigenvalue for a pair of matrices (A,B) is a scalar

* lambda or a ratio alpha/beta = lambda, such that A - lambda*B is

* singular. It is usually represented as the pair (alpha,beta), as

* there is a reasonable interpretation for beta=0, and even for both being zero.

*

* The right eigenvector v(j) corresponding to the eigenvalue

```

lambda(j)
* of (A,B) satisfies
*
*       $A * v(j) = \text{lambda}(j) * B * v(j).$ 
*
* The left eigenvector u(j) corresponding to the eigenvalue lambda(j)
* of (A,B) satisfies
*
*       $u(j)^{**H} * A = \text{lambda}(j) * u(j)^{**H} * B .$ 
*
* where u(j)**H is the conjugate-transpose of u(j).

```

On Sat, 27 Apr 2002 21:52:48 +0800, "tom" <tom2959@21cn.com> wrote:

```

> I found a matlab function for ellipse, but it is not easy for me translate
> to IDL. For example,
>
> % Solve eigensystem
> [gevec, geval] = eig(S,C);
>
> are there any function like eig(S,C) in IDL?
>
> The matlab for ellips fitting is as following, who have a idl version?
>
>
>
> -----
> -----
>
> function a = fitellipse(X,Y)
>
> % FITELLIPSE Least-squares fit of ellipse to 2D points.
> %      A = FITELLIPSE(X,Y) returns the parameters of the best-fit
> %      ellipse to 2D points (X,Y).
> %      The returned vector A contains the center, radii, and orientation
> %      of the ellipse, stored as (Cx, Cy, Rx, Ry, theta_radians)
> %
> % Authors: Andrew Fitzgibbon, Maurizio Pilu, Bob Fisher
> % Reference: "Direct Least Squares Fitting of Ellipses", IEEE T-PAMI, 1999
> %
> % This is a more bulletproof version than that in the paper, incorporating
> % scaling to reduce roundoff error, correction of behaviour when the input
> % data are on a perfect hyperbola, and returns the geometric parameters
> % of the ellipse, rather than the coefficients of the quadratic form.
> %
> % Example: Run fitellipse without any arguments to get a demo
> if nargin == 0
> % Create an ellipse

```

```

> t = linspace(0,2);
>
> Rx = 300
> Ry = 200
> Cx = 250
> Cy = 150
> Rotation = .4 % Radians
>
> x = Rx * cos(t);
> y = Ry * sin(t);
> nx = x*cos(Rotation)-y*sin(Rotation) + Cx;
> ny = x*sin(Rotation)+y*cos(Rotation) + Cy;
> % Draw it
> plot(nx,ny,'o');
> % Fit it
> fitellipse(nx,ny)
> % Note it returns (Rotation - pi/2) and swapped radii, this is fine.
> return
> end
>
> % normalize data
> mx = mean(X);
> my = mean(Y);
> sx = (max(X)-min(X))/2;
> sy = (max(Y)-min(Y))/2;
>
> x = (X-mx)/sx;
> y = (Y-my)/sy;
>
> % Force to column vectors
> x = x(:);
> y = y(:);
>
> % Build design matrix
> D = [ x.*x x.*y y.*y x y ones(size(x)) ];
>
> % Build scatter matrix
> S = D'*D;
>
> % Build 6x6 constraint matrix
> C(6,6) = 0; C(1,3) = -2; C(2,2) = 1; C(3,1) = -2;
>
> % Solve eigensystem
> [gevec, geval] = eig(S,C);
>
> % Find the negative eigenvalue
> I = find(real(diag(geval)) < 1e-8 & ~isinf(diag(geval)));
>

```

```

> % Extract eigenvector corresponding to negative eigenvalue
> A = real(gevec(:,l));
>
> % unnormalize
> par = [
>   A(1)*sy*sy, ...
>   A(2)*sx*sy, ...
>   A(3)*sx*sx, ...
>   -2*A(1)*sy*sy*mx - A(2)*sx*sy*my + A(4)*sx*sy*sy, ...
>   -A(2)*sx*sy*mx - 2*A(3)*sx*sx*my + A(5)*sx*sx*sy, ...
>   A(1)*sy*sy*mx*mx + A(2)*sx*sy*mx*my + A(3)*sx*sx*my*my ...
>   - A(4)*sx*sy*sy*mx - A(5)*sx*sx*sy*my ...
>   + A(6)*sx*sx*sy*sy ...
>   ]';
>
> % Convert to geometric radii, and centers
>
> thetarad = 0.5*atan2(par(2),par(1) - par(3));
> cost = cos(thetarad);
> sint = sin(thetarad);
> sin_squared = sint.*sint;
> cos_squared = cost.*cost;
> cos_sin = sint .* cost;
>
> Ao = par(6);
> Au = par(4) .* cost + par(5) .* sint;
> Av = - par(4) .* sint + par(5) .* cost;
> Auu = par(1) .* cos_squared + par(3) .* sin_squared + par(2) .* cos_sin;
> Avv = par(1) .* sin_squared + par(3) .* cos_squared - par(2) .* cos_sin;
>
> % ROTATED = [Ao Au Av Auu Avv]
>
> tuCentre = - Au./(2.*Auu);
> tvCentre = - Av./(2.*Avv);
> wCentre = Ao - Auu.*tuCentre.*tuCentre - Avv.*tvCentre.*tvCentre;
>
> uCentre = tuCentre .* cost - tvCentre .* sint;
> vCentre = tuCentre .* sint + tvCentre .* cost;
>
> Ru = -wCentre./Auu;
> Rv = -wCentre./Avv;
>
> Ru = sqrt(abs(Ru)).*sign(Ru);
> Rv = sqrt(abs(Rv)).*sign(Rv);
>
> a = [uCentre, vCentre, Ru, Rv, thetarad];
>
>

```


>
>

Subject: Re: Ellipse fitting?

Posted by [David Fanning](#) on Tue, 12 May 2009 21:18:03 GMT

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

robparker23@googlemail.com writes:

> I was hoping that someone could help with this.
>
> I have a satellite footprint which is more or less elliptical. I have
> the 4 coordinates of the top, bottom, left and right-most points and
> want to fit an ellipse to these 4 coordinates.
>
> I've put a diagram here to explain: <http://img217.imageshack.us/img217/5048/ellipse.png>
>
> I've googled this quite a bit and there seem to be lots of ways to fit
> ellipses to complicated data sets using techniques like optimal
> estimation but I can't find an easy/quick way of doing it. To do this
> manually I think I could just take the equation for an ellipse and
> adjust it to fit but I need to do this automatically for a large
> number of points where the shape of the ellipse will change depending
> on the location.
>
> This gets slightly more complicated (doesn't it always) as the shape
> may not quite be an ellipse due to the geometry of the satellite (it
> might have one side slightly "fatter" than the other) and it might be
> titled at an angle (so not horizontal).
>
> Does IDL have any ellipse fitting capabilities built in or is anyone
> familiar with a solution that might work?

You might try Fit_Ellipse:

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

The purpose of this program was to facilitate "blob analysis",
but you might be able to convert it to your purpose:

http://www.dfanning.com/ip_tips/blobanalysis.html

I guess a satellite footprint is more or less a blob. :-)

Cheers,

David

--

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

Subject: Re: Ellipse fitting?

Posted by [Kenneth P. Bowman](#) on Tue, 12 May 2009 22:10:28 GMT

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

In article

<36e1b1d7-1a57-4da6-afa4-d4925ed150ea@g20g2000vba.googlegroups.com>,
robarker23@googlemail.com wrote:

> Hi,
>
> I was hoping that someone could help with this.
>
> I have a satellite footprint which is more or less elliptical. I have
> the 4 coordinates of the top, bottom, left and right-most points and
> want to fit an ellipse to these 4 coordinates.
>
> I've put a diagram here to explain: <http://img217.imageshack.us/img217/5048/ellipse.png>
>
> I've googled this quite a bit and there seem to be lots of ways to fit
> ellipses to complicated data sets using techniques like optimal
> estimation but I can't find an easy/quick way of doing it. To do this
> manually I think I could just take the equation for an ellipse and
> adjust it to fit but I need to do this automatically for a large
> number of points where the shape of the ellipse will change depending
> on the location.
>
> This gets slightly more complicated (doesn't it always) as the shape
> may not quite be an ellipse due to the geometry of the satellite (it
> might have one side slightly "fatter" than the other) and it might be
> tilted at an angle (so not horizontal).
>
> Does IDL have any ellipse fitting capabilities built in or is anyone
> familar with a solution that might work?
>
> Cheers

If you know those four point, then you can find the center of the ellipse,
the semi-major axis a , and the semi-minor axis b , which are all of the parameters
of an ellipse (except for an arbitrary rotation).

Look up the definition of an ellipse in your analytical geometry text

or the fount of all wisdom

<http://en.wikipedia.org/wiki/Ellipse>

Ken Bowman

Subject: Re: Ellipse fitting?

Posted by [David Gell](#) on Wed, 13 May 2009 13:18:52 GMT

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

On May 12, 3:36 pm, robparke...@gmail.com wrote:

> Hi,
>
> I was hoping that someone could help with this.
>
> I have a satellite footprint which is more or less elliptical. I have
> the 4 coordinates of the top, bottom, left and right-most points and
> want to fit an ellipse to these 4 coordinates.
>
> I've put a diagram here to explain:<http://img217.imageshack.us/img217/5048/ellipse.png>
>
> I've googled this quite a bit and there seem to be lots of ways to fit
> ellipses to complicated data sets using techniques like optimal
> estimation but I can't find an easy/quick way of doing it. To do this
> manually I think I could just take the equation for an ellipse and
> adjust it to fit but I need to do this automatically for a large
> number of points where the shape of the ellipse will change depending
> on the location.
>
> This gets slightly more complicated (doesn't it always) as the shape
> may not quite be an ellipse due to the geometry of the satellite (it
> might have one side slightly "fatter" than the other) and it might be
> tilted at an angle (so not horizontal).
>
> Does IDL have any ellipse fitting capabilities built in or is anyone
> familiar with a solution that might work?
>
> Cheers

For what purpose do you want to fit the four points to an ellipse? If it is to plot the foot print, you might be better off computing the location of a number of points around the foot print and plot them. For reference see Wertz,J.R and W.J. Larson(eds), Space Mission Analysis and Design, Kluwer Academic Publishers, 1999 ISBN 1-881883-10-8. In particular Chapter 7 Section 2, "Earth Coverage"
