Convert position confidence ellipse to covariance matrix

Is there any way to compute a covariance matrix out of a confidence/uncertainty/error ellipse? I know how it's done the other way around, using a `2x2 covariance matrix` to compute an confidence ellipse (e.g. described here: http://www.visiondummy.com/2014/04/draw-error-ellipse-representing-covariance-matrix/).

Is this even possible or is necessary information missing?

My confidence ellipse is described by the length of both axis and the angle of ellipse rotation.

My approach so far: The axis lengths correspond to the two eigenvalues of the covariance matrix and defining the "spread". An ellipse angle of 0 means, there's no correlation between x & y. Covariance matrix without correlation

I created a new blank 2x2 matrix and assumed the angle is `zero` , e.g. I used the first eigenvalue and set it to `var_xx` . the same with the second eigenvalue and `var_yy` . Now I have a diagonal matrix, which describes the variance, but no rotation (correlation).

Now I used a 2D rotation matrix and the ellipse angle to rotate the previous created matrix.

> This approach seems wrong, because the matrix isn't symmetric anymore. Unfortunately a covariance matrix has to be symmetric.

Any ideas?

matrix ellipse uncertainty

edited Jan 23 at 14:13
Ganapathy
448 1 2 17

asked Jan 23 at 13:51
Thomas
1 2

1 Answer

It seems like I have a correct and working solution. The solution is derived from an answer in another forum: http://math.stackexchange.com/a/1119677

Matlab Example:

```
% ellipse param
longAxis = 20;
shortAxis = 10;
phi = 0;

% eigenvalues (this may vary from usecase to usecase)
eVal1 = longAxis;
eVal2 = shortAxis;

% compute eigenvectors
R = [cosd(phi), -sind(phi);
     sind(phi), cosd(phi)];
eVec1 = R * [eVal1; 0];
eVec2 = R * [0; eVal2];

% compute covariance matrix
% derived from: http://math.stackexchange.com/a/1119677
coVar = eVal1*(eVec1*eVec1')/(eVec1'*eVec1) + eVal2*(eVec2*eVec2')/(eVec2'*eVec2)
```

answered Jan 24 at 16:33

Thomas
1 2