

## Bivariate normal distribution fit to the image

When there is a correlation between horizontal and vertical orientation in our image, which is normal to observe in PITZ, the bivariate normal (Gaussian) distribution is a good choice to fit data instead of multiplication of two horizontal and vertical Gaussian distribution.

Bivariate normal distribution is 2-D multivariate normal distribution. The D-dimensional multivariate normal distribution is defined as below:

$$P(q; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^D |\Sigma|}} \exp\left(-\frac{1}{2}(q - \mu)^T \Sigma^{-1}(q - \mu)\right)$$

Which  $q$  is  $D \times 1$  position matrix,  $\mu$  is the center of the distribution and  $\Sigma$  is  $D \times D$  covariance matrix. For our special case,  $D=2$ , the equations becomes:

$$P(q; \mu, \Sigma) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(q - \mu)^T \Sigma^{-1}(q - \mu)\right)$$

$$q = \begin{pmatrix} x \\ y \end{pmatrix}, q - \mu = \begin{pmatrix} x' \\ y' \end{pmatrix}$$

I define  $\Sigma$  as below for the reason you will see soon.

$$\Sigma = \begin{bmatrix} C & -\frac{B}{2} \\ -\frac{B}{2} & A \end{bmatrix} \Rightarrow \Sigma^{-1} = \frac{1}{|\Sigma|} \begin{bmatrix} A & \frac{B}{2} \\ \frac{B}{2} & C \end{bmatrix} = \frac{1}{AC - \frac{B^2}{4}} \begin{bmatrix} A & \frac{B}{2} \\ \frac{B}{2} & C \end{bmatrix}$$

Now I rearrange the distribution by replacing  $q$  and  $\Sigma$  in the equation.

$$\begin{aligned} P(x', y'; \Sigma) &= \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2|\Sigma|} \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} A & \frac{B}{2} \\ \frac{B}{2} & C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}\right) \\ &= \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2|\Sigma|} (Ax^2 + Bxy + Cy^2)\right) \end{aligned}$$

$-\frac{1}{2|\Sigma|} (Ax^2 + Bxy + Cy^2) = -F''$  for the constant  $F''$  is the contour line of the distribution function.  $F''$  is simply defines in which  $\sigma$  the contour line should be located. Now if we rearrange it:

$$-\frac{1}{2|\Sigma|} (Ax^2 + Bxy + Cy^2) = -F'' = \frac{F}{2|\Sigma|} \Rightarrow Ax^2 + Bxy + Cy^2 + F = 0$$

It is interesting because it shows the ellipse equation.

The general form of ellipse equation is  $Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$  if  $B^2 - 4AC < 0$ . Without losing the generality if we moved the ellipse center to the origin,  $D$  and  $E$  becomes zero and the equation reduces to:

$$Ax^2 + Bxy + Cy^2 + F = 0 \text{ and } F > 0$$

This is exactly what we found for the contour lines equation and also it is the reason I define the  $\Sigma$  as above. Now we can build the  $M$  and  $M_0$  to find the ellipse parameters.

$$M = \begin{bmatrix} A & \frac{B}{2} \\ \frac{B}{2} & C \end{bmatrix} \& M_0 = \begin{bmatrix} F & \frac{D}{2} & \frac{E}{2} \\ \frac{D}{2} & A & \frac{B}{2} \\ \frac{E}{2} & \frac{B}{2} & C \end{bmatrix}$$

Then the ellipse parameters can be calculated by these equations:

$$a = \sqrt{-\frac{|M_0|}{|M|} \frac{1}{\lambda_1}} = \sqrt{\frac{F'}{\lambda_1}}$$

$$b = \sqrt{-\frac{|M_0|}{|M|} \frac{1}{\lambda_2}} = \sqrt{\frac{F'}{\lambda_2}}$$

$$\theta = \frac{1}{2} \text{atan2}(B, A - C)$$

a,b are the major and minor axis of ellipse respectively and  $\theta$  is the orientation of the major axis.

We have now all tools except we don't know how to find the  $\Sigma$  and  $\mu$  matrix from the image data.

They are calculated by these equations:

$$\begin{cases} \mu = \frac{\sum_{x,y} P(x,y) \times \begin{pmatrix} x \\ y \end{pmatrix}}{\sum_{x,y} P} \\ \Sigma = \frac{\sum_{x,y} P(x,y) \times \begin{pmatrix} x' & y' \end{pmatrix} \times \begin{pmatrix} x' \\ y' \end{pmatrix}}{\sum_{x,y} P} \\ \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} - \mu \end{cases}$$

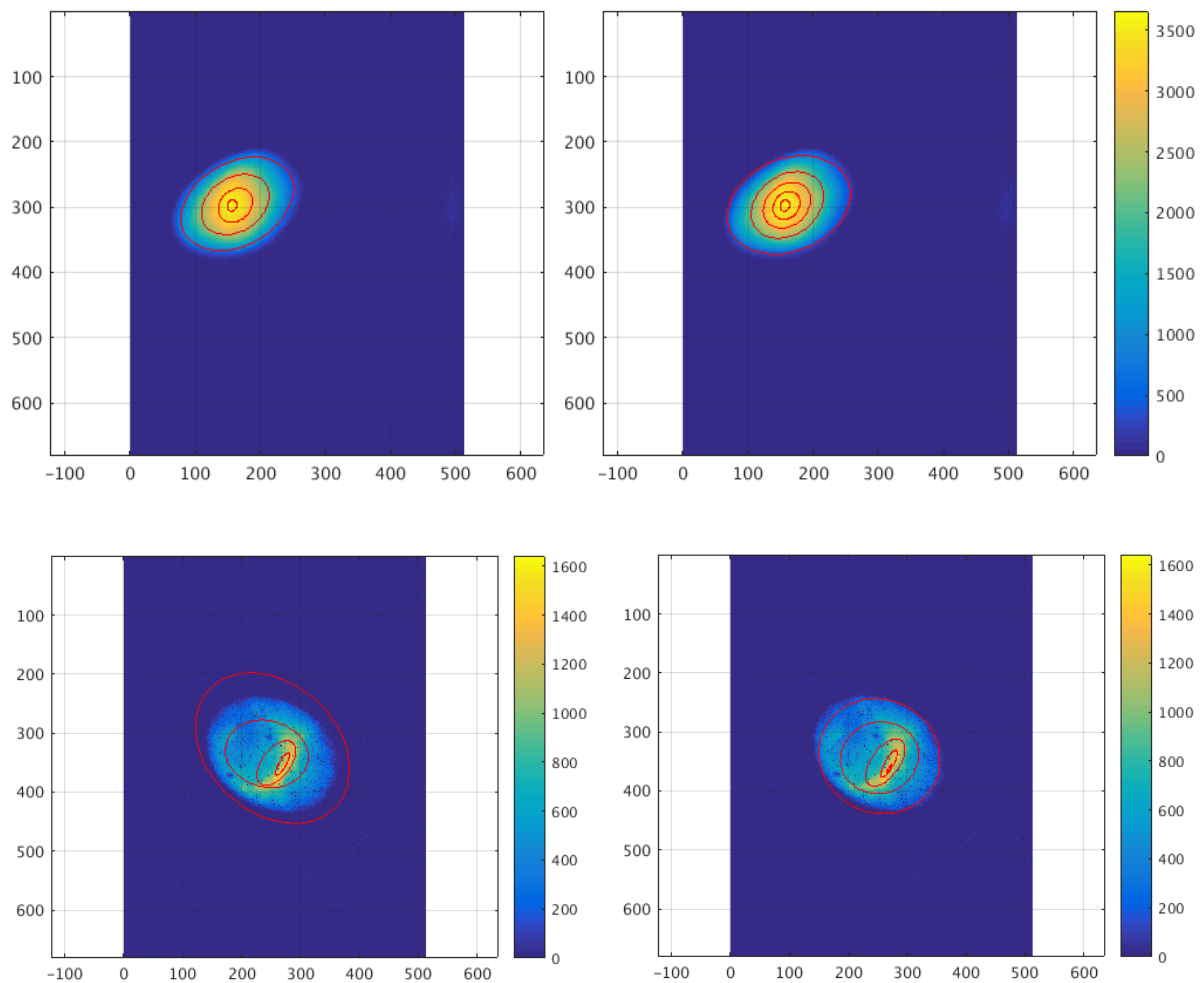
Now all of them can be transferred to a loop-free MATLAB function, the Matrix magic:

```
function [a,b,theta,mux,muy,asym,sigma]=bvn_ellipse(data,xscale,yscale,Fpp)
    [yl,xl]=size(data); % row->y column ->x
    p=data/sum(sum(data)); %normalized data
    x=(1:xl)*xscale; %Find the data center
    y=(1:yl)*yscale;
    mux=sum(p)*x';
    muy=sum(p')*y';
    xp=x-mux;
    yp=y-muy;
    sigma=zeros(2,2);
    sigma(1,1)=sum(p*(xp'.^2)); %Calculate the Sigma matrix
    sigma(2,2)=sum((yp.^2)*p);
    sigma(1,2)=yp*p*xp';
    sigma(2,1)=sigma(1,2);
    landa=eig(sigma); %Calculate the ellipse parameters
    Fp=2*det(sigma)*Fpp;
    a=sqrt(Fp/landa(1)); % major ellipse axis
    b=sqrt(Fp/landa(2)); % minor ellipse axis
    theta=180/pi*atan2(2*sigma(1,2),sigma(1,1)-sigma(2,2))/2; % orientation
    asym=landa(2)/landa(1)-1; % assymetry figure of merit = (a/b)^2-1
end
```

I attached this function with extra one function and two scripts. Now let look at them.

The 'normal\_distribution\_test\_generation\_fit.m' script file just an example which generate a bivariate distribution function and then fit the ellipse based on the 'bvn\_ellipse.m' function which I showed above. This script use the 'bvn\_gen.m' function to generate the distribution.

The 'imageloader\_MK\_BVN.m' script is another example. I used the script written by Mikhail which loads an image and reduces the background. I just add the bivariate normal distribution fit at the end. Here you can see some outputs from the script:



Here, In the left I sliced the data to [0.8,1], [0.6,0.8], [0.4,0.6], [0.2,0.4] of the maximum value and fit the ellipse for each of them. In the right I sliced the data to [1,0], [0.8,0], [0.6,0], [0.4,0], [0.2,0] of the maximum value. Just to be mention that the contour line is somewhere in this range and then it is not really shows the border. In the below image, you can see how the orientation and asymmetry can change from the center to the border then you should think which slice is more suitable for this kind of study. In the above image, more symmetric case, the elliptical parameters seems more or less identical.

I defined the  $(a/b)^2 - 1$  as the measure for asymmetry in the 'bvn\_ellipse.m' function. It should be zero for the symmetric case.

Feel free to use these codes.

Hamed

08.04.2019