

---

Subject: Re: Minimum area ellipse - quadratic optimisation?  
Posted by [Paolo Grigis](#) on Mon, 20 Feb 2006 17:41:46 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Olivia wrote:

>>> I posted on this topic before, but it is important that my ellipse  
>>> fitting method does not rely on convex hulls.  
>  
>  
>> Is there a special reason for this?  
>  
>  
> There is a special reason. I am doing a project on galaxy cluster  
> shapes, comparing the shapes of the clusters as determined by voronoi  
> tessellations, and by the minimum area ellipses. So fitting ellipses to  
> convex hulls would give a false comparison.  
>  
>  
>> Perhaps  
>> [http://geometryalgorithms.com/Archive/algorithm\\_0107/algorithm\\_0107.htm](http://geometryalgorithms.com/Archive/algorithm_0107/algorithm_0107.htm)  
>> could be of some help  
>  
>  
> This looks like just the thing I am after. I don't think finding the  
> center point and then finding the minimum area ellipse is a valid  
> method, after experimenting with it today. I understand the idea of  
> this fitting algorithm, but after reading the paper by Gaertner and  
> Schoenherr I doubt if I would be able to right a program to fit the  
> ellipses myself. Do you know of anyone who might have written one of  
> these types of programs for IDL? Thanks very much for your idea, and  
> help.  
>  
> Olivia  
>

Well, if you don't want to write any new code, you could just do a brute force optimization like this: suppose you have an efficient function which, given whatever ellipse-parametrization you use as input, tells you if all your points are inside the ellipse or not (you should have that by now). Define a new function of the ellipse parameters which returns the area of the ellipse if all points are inside and a large number if at least one point is outside. Then use a function minimizer (many exist, already implemented in IDL, any which does not rely too much on the function being smooth will do) to find the set of ellipse parameters which minimize your new function. If you feed it reasonable starting guesses it should converge relatively well and it might not be too slow. Maybe it is a bit of overkill, but if you really want to avoid implementing other algorithms...

Ciao,  
Paolo

---