Subject: Re: Pixel positions passing trough a curve
Posted by Helder Marchetto on Tue, 11 Oct 2016 14:01:25 GMT
View Forum Message <> Reply to Message

On Tuesday, October 11, 2016 at 3:06:01 PM UTC+2, Helder wrote:
> On Tuesday, October 11, 2016 at 2:36:44 PM UTC+2, miguelfigue...@gmail.com wrote:
>> Hi everyone,
>>
>> I am trying to construct a program that produces a position-velocity diagram whose shape can be fitted by a certain equation (and plotted).
>>
>> I wanted to know if it was possible to obtain somehow the pixels position trough a path I choose. For example, I draw a line (in ds9) from (0,0) to (10,0) and I want to know which pixels represent the line. In this case, the program would give (0,0), (1,0), (2,0),...,(10,0). The thing is that I want to draw an arbitrary path (not necesarily in ds9) and get automatically the points.
>>
>> For the moment, I put a circle, in ds9, in each pixel of the path and obtain the list of pixel positions and saving the region in xy format. I use this catalogue as an input of my program.
>> If there is a way to do it automatically (or some ideas) or an already existing program, I would like to know more.
>>
>> Thank you,
>> Miguel
>
> A counter-question: if going from (0,0) to (10,10), would you want:
> (a) (0,0), (1,1), (2,2),...,(9,9),(10,10)
> (b) (0,0), (1*cos(45),1*sin(45)), (2*cos(45),2*sin(45)), ...
>
> That said, as you can see, it's all about trigonometry and getting the details right. I've done this quite a few time using approach (b). If you wish, I can clean this up a bit and send it to your private email. Be aware that I've added a small degree of complexity by averaging along the perpendicular of x-pixel width (smoothens out the signal a bit).
>
> cheers, Helder

Ok, so I just put all the code online. Notice that I follow solution (b).
You can find the code in the post:
http://idl.marchetto.de/getting-line-profiles/

Let me know if you can recommend any improvements of any type or if you find any bugs. I haven't really made any checks to make sure the point are inside or are not too close to one another and so on...

Cheers,
Helder

PS: if you want to use a circle, you can obviously calculate the points as [2,n] array, however it is quite important to get the n value right, otherwise things don't work so well...