Subject: Re: Uneven Set of x y coordinates
Posted by Conor on Wed, 26 Sep 2007 13:30:54 GMT
View Forum Message <> Reply to Message

On Sep 25, 5:12 pm, "rpert...@gmail.com" <rpert...@gmail.com> wrote:
> Here is my problem:
> I have 2 sets of x,y coordinates:
>
> (20,3) (23,3)
> (50,3) (55,3)
> (70,3) (74,4)
> (110,3) (155,3)
> (150,4) (166,3)
> etc etc
> etc etc
> etc etc
>
> It is easy to see the first three x,y positions correspond (it is
> known these are close together, and also known that 2nd set is more to
> the right than 1st set, i.e. x2>x1)
>
> The fourth x,y inf first set however does not correspond to the 4th in
> 2nd set, in fact it is a surplus.
>
> I am therefore starting with my shortest array of coordinates and
> trying to match it with the appropriate coordinates in the long array
> discarding the excess points.
>
> So far I have used for, if and while loops...and I am able to identify
> which goes with which, but not able to reach the end of my goal: have
> identical size arrays of coordinates (smallest) with matching
> points...
>
> HELP!
> Thanks,
> RP

It sounds like you want a basic distance matching algorithm. i.e.
find the closest points in two lists that match to within a certain
tolerance. I do this a lot for matching star fields. If I have two
lists of star coordinates which ones are the same stars? Here's a
rather speedy program I use to do this (poorly commented).

http://astro.ufl.edu/~cmancone/pros/qfind.pro

The way it works is pretty simple. Let's say you have two star lists:

x1,y1 and x2,y2

you would call:

res = qfind(x1,y1,x2,y2,posshift=value)

Where the value you give posshift is how far two objects can be and still be a match (in your case, that looks like it should be about 6 or so). It returns an array of dimensions 2xn where n is the number of matching stars it found. Essentially, this is the equivelent of 2 where searches. The first column is indexes to use on the first star list, and the second column is the indexes to use on the second star list. So for instance:

```
x1 = [5,8,12,15]
y1 = [22,14,35,16]
x2 = [6,9,12,32]
y2 = [20,12,42,56]
res = qfind(x1,y1,x2,y2,posshift=5)
print,x1[res[0,*]],y1[res[0,*]]
print,x2[res[1,*]],y2[res[1,*]]
```

In this case the algorithm is designed so that it searches simply in position space - the objects don't have to be "physically" next to eachother in the arrays.
