
Subject: Re: x-y offsets

Posted by [Jeremy Bailin](#) on Thu, 20 May 2010 01:38:08 GMT

[View Forum Message](#) <> [Reply to Message](#)

On May 19, 7:45 pm, Gray <grayliketheco...@gmail.com> wrote:

> On May 19, 2:50 am, Jeremy Bailin <astroco...@gmail.com> wrote:

>

>

>

>

>

>> On May 18, 8:29 pm, Gray <grayliketheco...@gmail.com> wrote:

>

>>> Hi all,

>

>>> This is a variation on the 2D matching problem that I'm having trouble
>>> algorithm-ing (to coin an incredibly awkward word).

>

>>> I have two sets of XY coordinates of unequal length (i.e., $x_1/y_1/n_1$,
>>> $x_2/y_2/n_2$, $n_1 \neq n_2$). I want to find offsets in both X and Y that
>>> match the two sets as closely as possible (there will obviously be
>>> some unmatched coordinates in the larger set). I'm just looking for
>>> constant offsets, so basically (for $n_1 < n_2$) $x_1 + C_x \rightarrow x_2$, $y_1 + C_y \rightarrow$
>>> y_2 , with some elements of x_2 and y_2 being unmatched. How do I go
>>> about doing this? I don't think I can use JD's MATCH_2D because I
>>> don't know a priori what my matching radius is.

>

>>> Any suggestions? Thanks, as always!

>

>>> --Gray

>

>> I would be tempted to create a 2D histogram based on each set and then
>> cross-correlate them.

>

>> -Jeremy.

>

> How do you turn the cross-correlation into offsets? And, how do you
> intelligently choose a binsize for the histogram?

The first question is the easier one. ;-)

```
IDL> d = dist(5,5)
IDL> a = fltarr(25,25)
IDL> b = fltarr(25,25)
IDL> a[4,7] = d
IDL> b[0,0] = d
IDL> xcor = fft(/inverse, fft(a)*fft(b,/inverse))
IDL> maxcor = max(abs(xcor),loc)
```

```
IDL> print, array_indices(a,loc)
      4      7
```

Now, it's easy here because I know that there's one perfect matching location - it may be more ambiguous in a real situation (in which case you'll probably to assess the magnitude of all of the peaks within xcor to see if there are multiple plausible solutions). Also note that the answer wraps around - i.e. you should treat a value of 24 here as -1.

As for the binsize, it depends on your application. Ideally you would make the bins as small as the precision you expect to be able to achieve in determining the translational offset given your data (or even better, a factor of two smaller) - but if that means that your 2D histograms have one million bins in each direction then that won't work. ;-)

So in that case, I would go for a two-step process: in step 1, use the cross-correlation of the entire image using a coarse grid to get in the right ballpark. Then, if you think you should be good to within a length L, do a finer resolution cross-correlation just using a box of length L around each point (you might be able to ram the boxes all up against each other in a big image so you can do the cross-correlation of them all at once - never tried it).

-Jeremy.
