Subject: Re: Find shift between 2 star lists
Posted by Gray on Mon, 06 Dec 2010 23:24:46 GMT
View Forum Message <> Reply to Message

On Dec 6, 5:49 pm, Noah Johnson <noah.johnso...@gmail.com> wrote:

> Dear David,
>
> thank you for the response.
> I did read this article but it might not solve my problem.
> The problem is that one of the two fields is much more crowded than
> the other.
> For example if one of the two images is shifted 50-100 pixels when
> trying to find the closest star
> I might get a different star because there are so many stars that are
> really close to each other.
> That is why I was thinking that I should first try to find
> the shift between the images and then use a routine to find the
> closest match star.
> Is there such a routine?
>
> Thank you,
>
> Noah

Hi,

Take a look at this thread and see if it's kind of what you want:
http://groups.google.com/group/comp.lang.idl-pvwave/browse_t
hread/thread/f04e4da4993495f7/2c28002e6560d11d?q=#2c28002e65 60d11d

I have a (not at all documented) program that I wrote based on
Jeremy's solution. Basically, you pass it your two XY coordinate
lists and a binsize, and it outputs the optimal offset in x and y.
The binsize you choose depends on the size of your image (the x and y
ranges); I found that a binsize of 1.5 to 2 pixels (my matching radius
to match_2d was 1 pixel) worked fine and wasn't overly time-intensive. Calling sequence:

```
offsets = xy_offset(x1,y1,x2,y2,bin)
matches = match_2d(x1-offsets[0],y1-offsets[1],x2,y2,etc.)
```

```
FUNCTION xy_offset, x1, y1, x2, y2, bin
  on_error, 0
  xr = [min([x1,x2],max=tmp),tmp]+bin*[-1.,1.]
  yr = [min([y1,y2],max=tmp),tmp]+bin*[-1.,1.]
  hist1 = hist_2d(x1,y1,min1=xr[0],max1=xr[1],bin1=bin,$
    min2=yr[0],max2=yr[1],bin2=bin)
  hist2 = hist_2d(x2,y2,min1=xr[0],max1=xr[1],bin1=bin,$
    min2=yr[0],max2=yr[1],bin2=bin)
  hsz = size(hist1,/dimen)
  xcor=fft(/inverse,fft(hist1)*fft(hist2,/inverse))
  mxc = max(abs(xcor),lmxc)
  mxind = array_indices(hsz,lmxc,/dim)
  axcor = [[xcor,xcor,xcor],[xcor,xcor,xcor],[xcor,xcor,xcor]]
  bx = 7 & hbx = bx/2
  mxp = mxind + hsz
  aa = axcor[mxp[0]-hbx:mxp[0]+hbx,mxp[1]-hbx:mxp[1]+hbx]
  params = [0.,max(aa),1.,1.,hbx,hbx,0.]
  yfit=gauss2dfit(aa,params)
  refined = params[4:5]-hbx+mxind
  refined -= hsz * (refined gt hsz/2)
  offs = refined * bin
  ;check for reasonableness, or recurse with coarser binsize
  if (max(x1 - offs[0]) lt xr[0] or min(x1 - offs[0]) gt xr[1] or $
    max(y1 - offs[1]) lt yr[0] or min(y1 - offs[1]) gt yr[1]) then $
    return, xy_offset(x1,y1,x2,y2,bin*1.25)
  return, offs
end
```