
Subject: Re: Need Some Advice on Seperating Out Some Data

Posted by [rdellsy](#) on Mon, 14 Aug 2006 19:42:23 GMT

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

And now, without further ado, the final program. I hope the attributions granted are sufficient.

Feel free to tear it to shreds. =)

Thanks for all the help,

Rob

```
; D_AUTOSUM_1D
```

```
;-----
```

```
; FUNCTIONALITY:
```

```
; Type: Data Processing
```

```
; Categories: Expert Systems, Automation, Image Processing
```

```
; Step: Setting Cuts, Creating Sum Files
```

```
;-----
```

```
; AUTHORSHIP:
```

```
; Written by: Robert Dellsy Aug. 2006
```

```
; Modified from code written by JD Smith and posted to the IDL Usenet
```

```
; group: comp.lang.idl-pvwave on 8/9/06
```

```
; Modified from s_sum_1D and s_display_1D written by BX Cui and S  
Stratton
```

```
; for Rice Group.
```

```
; Version 0.2.0
```

```
;-----
```

```
; VERSION HISTORY:
```

```
; - 8/9/06: Program Started.
```

```
; VERSION INCREMENT: V0.0.0
```

```
; - 8/11/06: Modified cluster scalling by adding xsc, xep, and yep  
variables,
```

```
; added selection along a linear fit.
```

```
; VERSION INCREMENT: V0.1.0
```

```
; - 8/11/06: Added code from s_sum_1D to get the base data to play with  
and
```

```
; output, added ability to manually set cuts, added error  
detection.
```

```
; VERSION INCREMENT: v0.2.0
```

```
; - 8/14/06: Added code from s_display_1D to allow for displaying data,  
added
```

```
; disponly keyword functionality to allow for only displaying  
data,
```

```
; added comments and header.
```

```
;-----
```

```
; PURPOSE:
```

```
; This program is designed to replace s_display_1D and s_sum_1D as an  
expert
```

```
; system the creation of sum data files. It is therefore part of the
```

```

automation
; system to be used generally for all projects. In the short term, it
will work
; autonomously to ease the process of creating sum data files.
;-----
; REQUIRED PROGRAMS:
; - c_read_nih (BX Cui)
; - c_bpass (BX Cui)
; - c_feature (BX Cui)
;-----
; CALLING SEQUENCE:
; d_autosum_1d,mainfolder,movie,extent,separation
; Example:
; mainfoler='Users/SomeGuy/Desktop/Data/SourceFolder/'
; extent=11 ; Must be odd.
; separation=12 ; Must be even and greater than extent.
; xcuts=[0,640]
; ycuts=[0,200]
; ...
; ; Create sum data files
; cd,mainfolder
; mv=findfile('M*')
;
d_autosum_1d,mainfolder,mv(i),extent,separation,xcuts=xcuts, ycuts=ycuts
;-----
; KEYWORDS:
; - display - If set, the program will display an image from the movie
with
;   what it percieves as the sphere centers shown. If not set, then
;   that won't be displayed.
; - disponly - If set, the program will not produce a sum file, and
will only
;   loop through the data when it will display spheres. If not set,
the
;   program will still produce a sum data file even if 'display' is
set.
;   If set, also sets 'display'.
; - manual - If set, then manual cuts may be provided and will be acted
on.
;   If not set, then manual cuts will not be acted on even if
provided.
; - noauto - If set, then the autmatic, cluster-based detection will be
;   disabled. If not set, then automatic, cluster-based detection
is
;   enables. If set, also sets 'manual'.
;-----
; INPUTS:
; - mainfolder - STR - The directory being used.

```

```

; - extent - INT - This is the radius, in pixels, of a particle. This
needs
;   to be odd.
; - separation - INT - This is the minimal distance between the center
of one
;   particle, and the edge of another. This needs to be even and
greater
;   than extents.
; - cuts - FLTARR - This is a 2 element array containing information
for
;   removing false sphere centers. This only needs to be passed if
the
;   keywords 'manual' or 'noauto' are set. If either of these
keywords is
;   set, then 'cuts' must be passed.
;   [0] - This is the minimum brightness.
;   [1] - This is the maximum excentricity.
; - xcuts - FLTARR - The boundaries on the x-position of the sphere
centers.
; - ycuts - FLTARR - The boundaries on the y-position of the sphere
centers.
; - nframes - INT - The number of frames to be summed. If not set, then
the
;   entire movie will be summed.
; - masscut - FLTARR - The boundaries on the mass of a particle. Only
used
;   when keywords 'manual' or 'noauto' are set,
; - brightcut - FLTARR - The boundaries on the brightness of a
particle. Only
;   used when keywords 'manual' or 'noauto' are set.
; - freqdisp - INT - How often to display a plot. Only used when
keywords
;   'display' or 'disponly' are set.
; - nclust - INT - How many clusters to divide the data into.
; - yep - INT - The exponent that the y-coordinates are scaled by.
;-----
; VARIABLE SUMARY:
; - mv - FLTARR - A pixel-by-pixel, frame-by-frame conversion of the
movie into
;   an array.
; - i - Counter variable for FOR loop. Corresponds to the frame number.
; - c - FLTARR - The data from 'mv' for only one frame. Later, a
filtered
;   version of the same.
; - f - FLTARR - An array containing the data on the 'features', ie
possible
;   sphere centers from a frame.
;   [0] - The x-coordinate of the center.

```

```

;   [1] - The y-coordinate of the center.
;   [2] - The brightness of the particle.
;   [3] - The 'mass' of the particle.
;   [4] - The excentricity of the particle.
; - x - FLTARR - The x-position of the center of all the good spheres.
; - y - FLTARR - The y-position of the center of all the good spheres.
; - array - FLTARR - f([2,3])
; - w - FLTARR - The centers of gravity of each cluster of data.
; - cl - FLTARR - A listing of which cluster each sphere center belongs
in.
; - h - FLATARR - A histogram of 'cl'.
; - ri - FLTARR - The reverse indices of 'h'.
; - nh - INT - The number of elements in 'h'.
; - cen - FLTARR - The coordinates of the center of each cluster.
; - take - INTARR - A selection array.
; - lrc - INT - The subscript for the lower-right cluster.
; - keep1 - INTARR - A selection orray of all of the elements in the
lower-
;   right cluster.
; - fit - FLTARR - An array containing the y-intercept and slope of the
best-
;   fit line of the lower-right cluster.
; - bkmax - FLT - The maximum brightness of the lower-right cluster.
; - bkmin - FLT - The minimum brightness of the lower-right cluster.
; - bfit - FLTARR - A serries of brightness values for which the fit
line will
;   be calculated.
; - mfit - FLTARR - The calculated mass values of the fit line.
; - keep2 - INTARR - A selection array of spheres below the fit line
but not;
;   the good cluster.
; - keep - INTARR - A selection array of all good spheres.
; - dat - FLTARR - The values from 'f' of all the good spheres.
; - nparticles - FLTARR - The number of particles per frame.
; - iarray - INTARR - An array containing the frame number for each
particle.
; - sumi - FLTARR - An array of the data for the particle and their
frame
;   number.
; - result - FLTARR - An expansion of 'sumi' over all frames.
;-----
; TODO:
; - Work on the fitting and selection of non-cluster selected data.
; - Improve efficiency.
;-----
; LEGAL:
; Copyright 2006 Robert Dellsy, all rights reserved.
;

```

```

; This software is provided "as-is", without any express or implied
warranty.
; In no event will the authors be held liable for any damages arising
from
; the use of this software.
;
; Permission is granted to anyone to use this software for any
non-commercial
; purpose, and to modify and distribute the same, subject to the
following
; restrictions:
;
; 1. The origin of this software must not be misrepresented; you must
not
;   claim you wrote the original software. If you use this software in
a
;   product, an acknowledgment in the product documentation would be
;   appreciated, but is not required. In addition, then information
given
;   under the heading 'AUTHORSHIP' must be retained.
;
; 2. Altered source versions must be plainly marked as such, and must
not
;   be misrepresented as being the original software.
;
; 3. This notice may not be removed or altered from any source
distribution.
;
; 4. Otherwise, this code is covered by the Mozilla Public License 1.1
as
;   posted at http://www.opensource.org/licenses/mozilla1.1.php, with
the
;   understanding that where the license is different from this
notice, this
;   notice takes precedence.

```

```

; S1 - GENERAL PRE-RUN OPERATIONS

```

```

pro
d_autosum_1d,mainfolder,movie,extent,separation,cuts=cuts,xc uts=xcuts,$
  ycuts=ycuts,nframes=nframes,masscut=masscut,brightcut=bright cut,$
  display=display,disponly=disponly,freqdisp=freqdisp>manual=manual,$
noauto=noauto,nclust=nclust,yep=yep

```

```

if (keyword_set>manual) or keyword_set(noauto)) and not
keyword_set(cuts) then $

```

```

    print,'ERROR: Manual operation selected, and variable CUTS not
passed' & $
    return

```

```

tv1ct,[255,0,0,0,255,255],[0,255,0,255,255,0],[0,0,255,255,0,255],1

```

```

cd,mainfolder

```

```

if keyword_set(nclust) then nclust=nclust else nclust=4
if keyword_set(yep) then yep=yep else yep=4
xsc=1.
xep=1

```

```

if keyword_set(nframes) then $
  mv=c_read_nih(movie,first=0,last=nframes) $
else mv=c_read_nih(movie,/all)
nf=n_elements(mv(0,0,*))
;begin to process to get the sumdata
if keyword_set(xcuts) then xcuts=xcuts else xcuts=[0,640.]
if keyword_set(ycuts) then ycuts=ycuts else ycuts=[0,480.]
if keyword_set(masscut) then masscut=masscut else masscut=[0,1.e8]
if keyword_set(brightcut) then brightcut=brightcut else
brightcut=[0,1.e8]
nparticles=fltarr(nf)
if keyword_set(freqplot) then freqplot=freqplot else freqplot=1000

```

```

; S2 - GET FEATURES

```

```

if keyword_set(display) then window,20,xsize=800,ysize=800
for i=0,nf-1 do begin
  if keyword_set(disponly) and not (i mod freqdisp) eq 0 then
continue
  c=reform(mv(*,*,i))
  c=c_bpass(c,1,extent)
  f=c_feature(c,separation-1.,separation)

```

```

; S3 - BACKWARDS COMPATABILITY

```

```

; This section allows for manually setting cuts.
if keyword_set(manual) or keyword_set(noauto) then do begin
  w=where(f(2,*) gt cuts(0) and f(4,*) le cuts(1) and $
    f(0,*) gt xcuts(0) and f(0,*) le xcuts(1) and $
    f(1,*) gt ycuts(0) and f(1,*) le ycuts(1))
  f=f(*,w)
  w=where(f(3,*) gt masscut[0] and f(3,*) le masscut[1] and $
    f(2,*) gt brightcut[0] and f(2,*) le
brightcut[1],count)
  f=f(*,w)

```

```

endif

if keyword_set(nauto) and (keyword_set(display) or $
keyword_set(disponly)) and (i mod freqdisp) eq 0 then do begin
    print,'the number of spheres found for frame',i,' is
',n_elements(f(0,*))
    x=reform(f(0,*))
    y=reform(f(1,*))
    dgcont_image,bytescl(mv(*,*,i)),/nocont,title='frame'+string( j)
    oplot,x,y,psym=2,color=255
endif
if keyword_set(nauto) then goto,writedata

```

; S4 - AUTOMATICALLY FIND GOOD SPHERE CENTERS

```

array=[f(2,*),f(3,*)]
array[0,*]=(xsc*array[0,*])^xep
array[1,*]=array[1,*]^yep
w=clust_wts(array,N_CLUSTERS=nclust)

window,0
!p.multi=[0,1,1]
plot,f[2],f[3],psym=2
oplot,(w[0,;]/xsc),w[1,*],psym=5,symsize=2.5

cl=cluster(array,w)
h=histogram(cl,REVERSE_INDICES=ri)
nh=n_elements(h)

cen=make_array(2,nh,VALUE=!VALUES.F_NAN)
for i=0,nh-1 do begin
    if ri[i+1] eq ri[i] then continue
    take=ri[ri[i]:ri[i+1]-1]
    oplot,f[2,take],f[2,take],PSYM=4,COLOR=i+1
    cen[0,i]=[mean(f[2,take]),mean(f[3,take])]
endfor

;; Find the lower right cluster
void=max(cen[0,*]-cen[1,*],lrc,/NAN)

;; Highlight it
keep1=ri[ri[lrc]:ri[lrc+1]-1]

fit=linfit(f[2,keep1],f[3,keep1])
bkmax=max(f[2,keep1],min=bkmin)
bfit=[1,[bkmax+1]/4,[bkmax+1]/2,3*[bkmax+1]/4,bkmax]
mfit=(fit[1]*bfit)+fit[0]

```

```
oplot,bfit,mfit
```

```
;xhist=histogram(bm[0,*],min=0,max=xkmin,binsize=100)
;print,xhist
;window,1
;plot,xhist[0,*],xhist[1,*]
```

```
keep2=where(f[3,*] le (fit[1]*f[2,*]+fit[0]) and $
  f[2,*] ge xkmin/4 and f[2,*] le xkmin and $
  f[0,*] gt xcuts[0] and f[0,*] le xcuts[1] and $
  f[1,*] gt ycuts[0] and f[1,*] le ycuts[1])
```

```
keep=[keep2,keep1]
dat=[f(*,keep)]
oplot,dat[2],dat[3],PSYM=6,SYMSIZE=2
```

```
; S4 - WRITING DATA
```

```
writedata:
```

```
nparticles[i]=count
iarray=fltarr(1,count)+i
sumi=[dat,iarray]
if (i eq 0) then result=sumi $
else result=[[result],[sumi]]
```

```
if keyword_set(display) or keyword_set(disponly) and $
(i mod freqdisp) eq 0 then do begin
  print,'the number of spheres found for frame',i,' is
',n_elements(f(0,*))
  x=reform(f(0,*))
  y=reform(f(1,*))
  dgcont_image,bytescl(mv(*,*,i)),/nocont,title='frame'+string( j)
  oplot,x,y,psym=2,color=255
endif
endfor
```

```
write_gdf,result,'sumdata_'+movie
write_gdf,nparticles,'nparticles_'+movie
```

```
end
```
