

---

Subject: Need Some Advice on Separating Out Some Data  
Posted by [rdellsy](#) on Tue, 08 Aug 2006 19:13:19 GMT

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

---

<http://photos1.blogger.com/blogger/4016/2263/320/graphroi.png>

The above is a plot of my data (minus the red polygon). I need to separate the data inside the red polygon (real data) from the data outside the red polygon (noise, for lack of a better term). All of these points are already contained in an array. I'm just trying to figure out a way for the computer to automatically figure out what is noise and what isn't based on that plot distribution. Each data set is slightly different, but has the same overall distribution, and, for properly dialed in data, there is always that characteristic separation between the good stuff and the bad stuff. Currently, we are manually setting x-boundaries and y-boundaries on our data.

Thanks in advance,  
Rob

---

---

Subject: Re: Need Some Advice on Separating Out Some Data  
Posted by [JD Smith](#) on Wed, 09 Aug 2006 22:17:18 GMT

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

---

On Wed, 09 Aug 2006 13:13:12 -0700, rdellsy wrote:

> Thanks for that. I took it, and played around with it a bit to get it  
> to work. [Errors I found were: x and y don't concatenate in the line  
> 'array=transpose([[x],[y]])' and I found I had to comment away the  
> /ISOTROPIC in the plotting.] Unfortunately, it seems that cluster  
> separates on a purely 1 dimensional basis. I tried discarding the  
> histogram related code in favor of a much simpler system in case that  
> was the problem, and it still didn't work. If you look at the data set  
> I provided, the problem should be self evident.

Probably your x,y are column vectors. I can't parse that data set;  
please repost in plain ASCII. I'm not sure why you say it works  
1-dimensionally. Did you try the example as given with the fake cluster  
data?

> Incidentally, I replaced everything from  
> h=histogram(c,reverse\_indices=ri) down to the second to last line with:  
> --  
> plot,x,y,psym=2  
> bmax=max(array[0,\*],maxsubsc)  
> goodc=c[maxsubsc]  
> keep=where(c[\*] eq goodc)  
> --

- > I feel that my code may be a tad more efficient, though I don't know
- > how efficient the WHERE command is.

HISTOGRAM is more efficient than WHERE, but then again if it's not slowing you down, it's a bit harder to parse, and you're only searching on a few cluster index values. You don't need c[\*] above: that just slows things down unnecessarily.

- > Anywho, I'm looking CLUSTER\_TREE right now, which shows some more
- > promise. If I understand it correctly, it works using distance apart,
- > not coordinates which is a bit more useful, I think, for my problem.
- > I'm just not sure how I can take the output and turn it into a set of
- > clusters.

I think CLUSTER does similar, it just doesn't build a "tree" of cluster membership.

JD

---

---

Subject: Re: Need Some Advice on Separating Out Some Data  
Posted by [rdellsy](#) on Thu, 10 Aug 2006 18:44:56 GMT

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

---

With the generated numbers, it seemed to work fine. Here is a comma delimited version (.csv) of my data:

<http://s2.quicksharing.com/v/6325147/bm.csv.html>

With that and Excel (or your spreadsheet application of choice), you should be able to get just about any data format out of that.

Thanks,  
Rob

JD Smith wrote:

- > On Wed, 09 Aug 2006 13:13:12 -0700, rdellsy wrote:
- >
- >> Thanks for that. I took it, and played around with it a bit to get it
- >> to work. [Errors I found were: x and y don't concatenate in the line
- >> 'array=transpose([[x],[y]])' and I found I had to comment away the
- >> /ISOTROPIC in the plotting.) Unfortunately, it seems that cluster
- >> separates on a purely 1 dimensional basis. I tried discarding the
- >> histogram related code in favor of a much simpler system in case that
- >> was the problem, and it still didn't work. If you look at the data set
- >> I provided, the problem should be self evident.
- >
- > Probably your x,y are column vectors. I can't parse that data set;
- > please repost in plain ASCII. I'm not sure why you say it works
- > 1-dimensionally. Did you try the example as given with the fake cluster

> data?  
>  
>> Incidentally, I replaced everything from  
>> h=histogram(c,reverse\_indices=ri) down to the second to last line with:  
>> --  
>> plot,x,y,psym=2  
>> bmax=max(array[0,\*],maxsubsc)  
>> goodc=c[maxsubsc]  
>> keep=where(c[\*] eq goodc)  
>> --  
>> I feel that my code may be a tad more efficient, though I don't know  
>> how efficient the WHERE command is.  
>  
> HISTOGRAM is more efficient than WHERE, but then again if it's not slowing  
> you down, it's a bit harder to parse, and you're only searching on a few  
> cluster index values. You don't need c[\*] above: that just slows things  
> down unnecessarily.  
>  
>> Anywho, I'm looking CLUSTER\_TREE right now, which shows some more  
>> promise. If I understand it correctly, it works using distance apart,  
>> not coordinates which is a bit more useful, I think, for my problem.  
>> I'm just not sure how I can take the output and turn it into a set of  
>> clusters.  
>  
> I think CLUSTER does similar, it just doesn't build a "tree" of  
> cluster membership.  
>  
> JD

---

---

Subject: Re: Need Some Advice on Seperating Out Some Data  
Posted by [rdellsy](#) on Thu, 10 Aug 2006 19:57:19 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

We use read\_gdf() to read it. If you have read\_gdf() already, then it's  
variableToContainData=read\_gdf(filename), where filename is a string  
containing the filename with the extension .gdf. I also posted a comma  
delimited version of the same data. Take that into Excel (it should  
open as an Excel document), then Save As and you can export it as any  
Excel data type.

- Rob

kuyper@wizard.net wrote:

> rdellsy@gmail.com wrote:  
>> <http://s8.quicksharing.com/v/2874818/bm.gdf.html>  
>  
> I don't recognise that file type. How do I read it? I'd prefer an  
> answer in terms of IDL commands.

---

Subject: Re: Need Some Advice on Seperating Out Some Data  
Posted by [JD Smith](#) on Thu, 10 Aug 2006 22:10:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Thu, 10 Aug 2006 11:44:56 -0700, rdellsy wrote:

> With the generated numbers, it seemd to work fine. Here is a comma  
> delimited version (.csv) of my data:  
> <http://s2.quicksharing.com/v/6325147/bm.csv.html>  
> With that and Excel (or your speadsheet application of choice), you  
> should be able to get just about any data format out of that.  
> Thanks,  
> Rob

Aha, right you are... it appears CLUSTER likes roundish, not elongated clusters. I also couldn't get it to select your clump.

Good luck,

JD

---

---

Subject: Re: Need Some Advice on Seperating Out Some Data  
Posted by [rdellsy](#) on Fri, 11 Aug 2006 05:39:50 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

I'm working on doing a cluster tree and getting say the lower-right cluster and the one or two nearest neighbors (sp?). I may still be loosing some data though. Another possibilty would be compressing the data, say, by half, and see if that helps.

Thanks,  
Rob

JD Smith wrote:

> On Thu, 10 Aug 2006 11:44:56 -0700, rdellsy wrote:  
>  
>> With the generated numbers, it seemd to work fine. Here is a comma  
>> delimited version (.csv) of my data:  
>> <http://s2.quicksharing.com/v/6325147/bm.csv.html>  
>> With that and Excel (or your speadsheet application of choice), you  
>> should be able to get just about any data format out of that.  
>> Thanks,  
>> Rob  
>  
> Aha, right you are... it appears CLUSTER likes roundish, not elongated  
> clusters. I also couldn't get it to select your clump.  
>  
> Good luck,

>  
> JD

---

---

Subject: Re: Need Some Advice on Seperating Out Some Data  
Posted by [James Kuyper](#) on Fri, 11 Aug 2006 17:22:35 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

rdellsy@gmail.com wrote:

> I'm working on doing a cluster tree and getting say the lower-right  
> cluster and the one or two nearest neighbors (sp?). I may still be  
> loosing some data though. Another possibilty would be compressing the  
> data, say, by half, and see if that helps.  
> Thanks,  
> Rob

```
IDL> help,data  
DATA          FLOAT    = Array[2, 681]
```

If all of the dimensions of your data have the same physical meaning,  
then  
you don't need to do anything to your data. However, I got the  
following  
results:

```
IDL> print,stddev(data[0,*]),stddev(data[1,*])  
2748.5689    1.7135388
```

Which implies to me that your x and y coordinates probably have  
drastically  
different meanings, so they need to be scaled to have a meaningful  
distance  
measurement. The simplest way is to base the scale factors on the  
standard deviations:

```
IDL> scaled = data  
IDL> scaled[0,*] /= stddev(data[0,*])  
IDL> scaled[1,*] /= stddev(data[1,*])
```

I recommend, since you're analyzing many different but comparable  
datasets, to use a single scaling factor on each axis for all the  
datasets; otherwise it will be difficult to compare your results  
between one dataset and another.

```
IDL> pairedistance = DISTANCE_MEASURE(scaled)  
IDL> clusters =  
CLUSTER_TREE(pairedistance,linkdistance,LINKAGE=0,data=scaled )
```

I'm surprised by the fact that I haven't been able to locate an IDL function or procedure for taking the output from CLUSTER\_TREE and using it to determine cluster membership at the point when there are N clusters left, so I wrote my own:

```
FUNCTION cluster_member, clusters
  dims = SIZE(clusters, /DIMENSIONS)
  num = dims[1] + 1
  membership = INTARR(num, num-1)
  work = indgen(num)
  FOR i=0, num-2 DO BEGIN
    newclust = WHERE (work eq clusters[0,i] OR work EQ
clusters[1,i])
    work[newclust] = num+i
    membership[0,i] = work
  ENDFOR

  RETURN, membership
END
```

There's probably a more efficient way of handling that loop. The row membership[\* ,0] identifies num-1 different clusters; membership[\* ,1] identifies num-2 different clusters; etc.

```
IDL> membership = cluster_member(clusters)
```

To get the results where everything's been merged into three clusters, look at membership[\* ,679]:

```
IDL> print, membership[uniq(membership[* ,677],
sort(membership[* ,677])),677]
  1341  1357  1358
IDL> plot,data[0,*],data[1,*],psym=3
IDL> c1341 = WHERE(membership[* ,677] eq 1341)
IDL> c1357 = WHERE(membership[* ,677] eq 1357)
IDL> c1358 = WHERE(membership[* ,677] eq 1358)
IDL> oplot,data[0,c1358],data[1,c1358],PSYM=2
```

Which is, I think, is precisely the cluster you're looking for.

---

Subject: Re: Need Some Advice on Seperating Out Some Data  
Posted by [JD Smith](#) on Fri, 11 Aug 2006 18:44:35 GMT  
[View Forum Message](#) <> [Reply to Message](#)

On Fri, 11 Aug 2006 10:22:35 -0700, kuyper wrote:

```

> rdellsy@gmail.com wrote:
>> I'm working on doing a cluster tree and getting say the lower-right
>> cluster and the one or two nearest neighbors (sp?). I may still be
>> loosing some data though. Another possibility would be compressing the
>> data, say, by half, and see if that helps.
>> Thanks,
>> Rob
>
> IDL> help,data
> DATA          FLOAT    = Array[2, 681]
>
> If all of the dimensions of your data have the same physical meaning,
> then
> you don't need to do anything to your data. However, I got the
> following
> results:
>
> IDL> print,stddev(data[0,*]),stddev(data[1,*])
>    2748.5689    1.7135388
>
> Which implies to me that your x and y coordinates probably have
> drastically
> different meanings, so they need to be scaled to have a meaningful
> distance
> measurement. The simplest way is to base the scale factors on the
> standard deviations:
>
> IDL> scaled = data
> IDL> scaled[0,*] /= stddev(data[0,*])
> IDL> scaled[1,*] /= stddev(data[1,*])
>
> I recommend, since you're analyzing many different but comparable
> datasets, to use a single scaling factor on each axis for all the
> datasets; otherwise it will be difficult to compare your results
> between one dataset and another.
>
> IDL> pairdistance = DISTANCE_MEASURE(scaled)
> IDL> clusters =
> CLUSTER_TREE(pairdistance,linkdistance,LINKAGE=0,data=scaled )
>
> I'm surprised by the fact that I haven't been able to locate an IDL
> function or procedure for taking the output from CLUSTER_TREE and using
> it to determine cluster membership at the point
> when there are N clusters left, so I wrote my own:
>
> FUNCTION cluster_member, clusters
>   dims = SIZE(clusters, /DIMENSIONS)

```

```
> num = dims[1] + 1
> membership = INTARR(num, num-1)
> work = indgen(num)
> FOR i=0, num-2 DO BEGIN
>     newclust = WHERE (work eq clusters[0,i] OR work EQ
> clusters[1,i])
>     work[newclust] = num+i
>     membership[0,i] = work
> ENDFOR
>
> RETURN, membership
> END
>
> There's probably a more efficient way of handling that loop.
```

Very cool! I'll have to remember this one. If you only care about n remaining clusters, you can simplify somewhat to:

```
function cluster_member, clusters,n
  dims = SIZE(clusters, /DIMENSIONS)
  num = dims[1] + 1L
  n>=1
  work = lindgen(num)
  for i=0L, num-1L-n do $
    work[where(work eq clusters[0,i] OR work eq clusters[1,i])]= num+i
  return, work
end
```

JD

---

---

Subject: Re: Need Some Advice on Seperating Out Some Data  
Posted by [rdellsy](#) on Fri, 11 Aug 2006 19:43:57 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Interesting. I found that if I take the 4th power of the y variable, I get a few decent clusters. I was planning on doing a linear fit of those good clusters, and then using that to find the trailing data. Add in a small histogram to find the lower x bound of what I want and it should work. Your idea also sounds promissing. I'm getting close. Hopefully, I'll have something that'll work by Tuesday, and I'll let you all tear it to pieces...err find more efficient ways of doing it.  
=)  
Thanks,  
- Rob

JD Smith wrote:



> On Fri, 11 Aug 2006 10:22:35 -0700, kuyper wrote:

>

>> rdellsy@gmail.com wrote:

>>> I'm working on doing a cluster tree and getting say the lower-right

>>> cluster and the one or two nearest neighbors (sp?). I may still be

>>> loosing some data though. Another possibility would be compressing the

>>> data, say, by half, and see if that helps.

>>> Thanks,

>>> Rob

>>

>> IDL> help,data

>> DATA        FLOAT    = Array[2, 681]

>>

>> If all of the dimensions of your data have the same physical meaning,

>> then

>> you don't need to do anything to your data. However, I got the

>> following

>> results:

>>

>> IDL> print,stddev(data[0,\*]),stddev(data[1,\*])

>>     2748.5689     1.7135388

>>

>> Which implies to me that your x and y coordinates probably have

>> drastically

>> different meanings, so they need to be scaled to have a meaningful

>> distance

>> measurement. The simplest way is to base the scale factors on the

>> standard deviations:

>>

>> IDL> scaled = data

>> IDL> scaled[0,\*] /= stddev(data[0,\*])

>> IDL> scaled[1,\*] /= stddev(data[1,\*])

>>

>> I recommend, since you're analyzing many different but comparable

>> datasets, to use a single scaling factor on each axis for all the

>> datasets; otherwise it will be difficult to compare your results

>> between one dataset and another.

>>

>> IDL> pairdistance = DISTANCE\_MEASURE(scaled)

>> IDL> clusters =

>> CLUSTER\_TREE(pairdistance,linkdistance,LINKAGE=0,data=scaled )

>>

>> I'm surprised by the fact that I haven't been able to locate an IDL

>> function or procedure for taking the output from CLUSTER\_TREE and using

>> it to determine cluster membership at the point

>> when there are N clusters left, so I wrote my own:

>>

>> FUNCTION cluster\_member, clusters

```

>>  dims = SIZE(clusters, /DIMENSIONS)
>>  num = dims[1] + 1
>>  membership = INTARR(num, num-1)
>>  work = indgen(num)
>>  FOR i=0, num-2 DO BEGIN
>>      newclust = WHERE (work eq clusters[0,i] OR work EQ
>> clusters[1,i])
>>      work[newclust] = num+i
>>      membership[0,i] = work
>>  ENDFOR
>>
>>  RETURN, membership
>> END
>>
>> There's probably a more efficient way of handling that loop.
>
> Very cool! I'll have to remember this one. If you only care about n
> remaining clusters, you can simplify somewhat to:
>
> function cluster_member, clusters,n
>  dims = SIZE(clusters, /DIMENSIONS)
>  num = dims[1] + 1L
>  n>=1
>  work = lindgen(num)
>  for i=0L, num-1L-n do $
>    work[where(work eq clusters[0,i] OR work eq clusters[1,i])]= num+i
>  return, work
> end
>
> JD

```

---

Subject: Re: Need Some Advice on Seperating Out Some Data  
 Posted by [rdellsy](#) on Fri, 11 Aug 2006 20:44:22 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Sorry for the double post.  
 Okay, I'm in the home stretch here. I'm just trying to figure out how  
 to use a WHERE command or a histogram to select only the data that lies  
 below a given line specified in  $y=mx+b$  form.  
 Thanks for all ya'll's help.  
 - Rob

rdellsy@gmail.com wrote:  
 > Interesting. I found that if I take the 4th power of the y variable, I  
 > get a few decent clusters. I was planning on doing a linear fit of  
 > those good clusters, and then using that to find the trailing data. Add  
 > in a small histogram to find the lower x bound of what I want and it

> should work. Your idea also sounds promising. I'm getting close.  
> Hopefully, I'll have something that'll work by Tuesday, and I'll let  
> you all tear it to pieces...err find more efficient ways of doing it.  
> =)  
> Thanks,  
> - Rob  
>  
>  
> JD Smith wrote:  
>> On Fri, 11 Aug 2006 10:22:35 -0700, kuyper wrote:  
>>  
>>> rdellsy@gmail.com wrote:  
>>>> I'm working on doing a cluster tree and getting say the lower-right  
>>>> cluster and the one or two nearest neighbors (sp?). I may still be  
>>>> loosing some data though. Another possibilty would be compressing the  
>>>> data, say, by half, and see if that helps.  
>>>> Thanks,  
>>>> Rob  
>>>  
>>> IDL> help,data  
>>> DATA        FLOAT    = Array[2, 681]  
>>>  
>>> If all of the dimensions of your data have the same physical meaning,  
>>> then  
>>> you don't need to do anything to your data. However, I got the  
>>> following  
>>> results:  
>>>  
>>> IDL> print,stddev(data[0,\*]),stddev(data[1,\*])  
>>>       2748.5689     1.7135388  
>>>  
>>> Which implies to me that your x and y coordinates probably have  
>>> drastically  
>>> different meanings, so they need to be scaled to have a meaningful  
>>> distance  
>>> measurement. The simplest way is to base the scale factors on the  
>>> standard deviations:  
>>>  
>>> IDL> scaled = data  
>>> IDL> scaled[0,\*] /= stddev(data[0,\*])  
>>> IDL> scaled[1,\*] /= stddev(data[1,\*])  
>>>  
>>> I recommend, since you're analyzing many different but comparable  
>>> datasets, to use a single scaling factor on each axis for all the  
>>> datasets; otherwise it will be difficult to compare your results  
>>> between one dataset and another.  
>>>  
>>> IDL> pairedistance = DISTANCE\_MEASURE(scaled)

```

>>> IDL> clusters =
>>> CLUSTER_TREE(pairdistance,linkdistance,LINKAGE=0,data=scaled )
>>>
>>> I'm surprised by the fact that I haven't been able to locate an IDL
>>> function or procedure for taking the output from CLUSTER_TREE and using
>>> it to determine cluster membership at the point
>>> when there are N clusters left, so I wrote my own:
>>>
>>> FUNCTION cluster_member, clusters
>>>   dims = SIZE(clusters, /DIMENSIONS)
>>>   num = dims[1] + 1
>>>   membership = INTARR(num, num-1)
>>>   work = indgen(num)
>>>   FOR i=0, num-2 DO BEGIN
>>>     newclust = WHERE (work eq clusters[0,i] OR work EQ
>>> clusters[1,i])
>>>     work[newclust] = num+i
>>>     membership[0,i] = work
>>>   ENDFOR
>>>
>>>   RETURN, membership
>>> END
>>>
>>> There's probably a more efficient way of handling that loop.
>>
>> Very cool! I'll have to remember this one. If you only care about n
>> remaining clusters, you can simplify somewhat to:
>>
>> function cluster_member, clusters,n
>>   dims = SIZE(clusters, /DIMENSIONS)
>>   num = dims[1] + 1L
>>   n>=1
>>   work = lindgen(num)
>>   for i=0L, num-1L-n do $
>>     work[where(work eq clusters[0,i] OR work eq clusters[1,i])]= num+i
>>   return, work
>> end
>>
>> JD

```

---

Subject: Re: Need Some Advice on Seperating Out Some Data  
 Posted by [James Kuyper](#) on Fri, 11 Aug 2006 20:48:49 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

rdel...@gmail.com wrote:  
 > Sorry for the double post.  
 > Okay, I'm in the home stretch here. I'm just trying to figure out how

- > to use a WHERE command or a histogram to select only the data that lies
- > below a given line specified in  $y=mx+b$  form.
- > Thanks for all ya'll's help.

below = WHERE( y lt m\*x+b)

---

---

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_iD 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 automatic, 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 SUMMARY:
; - 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,xcuts=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

```

```

tvlct,[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(noauto) 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,bytsc1(mv(*,*,i)),/nocont,title='frame'+string( j)
    oplot,x,y,psym=2,color=255
endif
if keyword_set(noauto) 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,bytsc1(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
```

---