Subject: Re: tracking clusters through multiple timesteps
Posted by David Fanning on Mon, 20 Feb 2012 17:49:49 GMT

Ian_Ashpole writes:

> I have a binary dataset (flags of cloud presence or absence) and i
> would like to find a way of tracking clusters of 1's through time,
> with an aim of finding the start point (in space and time) of the
> object, the end point, its maximum size and its trajectory. I have
> experience of using 'label_region' to identify individual objects
> within an image by pixel connectivity, but i am completely at a loss
> for correct way to move forward from here.
> I am dealing with an array of dimensions 1133,751,8832, with 8832
> being the number of timesteps i ultimately aim to work through.
> Any help, tips or advice would be hugely appreciated!

Yikes! Clouds can get bigger or smaller, blow around
to hither and gone, split apart, come together, and
do all manner of assumption sundering things. They
are Shape-shifters, right?

If it were me, I'd be thinking about going fishing
rather than working on this. :-)

Cheers,

David

P.S. Donno. If the time sequences are close
enough together I guess you could cobble something
together that looked at overlap from one image to
the next. But, I can easily think of about a thousand
things that might go wrong with this approach. :-)

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: tracking clusters through multiple timesteps
Posted by Ian[1] on Mon, 20 Feb 2012 18:20:06 GMT

Haha, thanks for such a quick reply David - i really wish i could go fishing instead, but unfortunately this ugly problem needs dealing with first :(The time resolution is very high - 15 minutes - so there is a huge degree of overlap between blob at t and blob at t+1.

For now i would like to work on an idealised case with no splitting or merging - but where the cloud will move quite a way from its start to finish location. I should specify that i am very much an IDL novice but it seems logical to pass the index given to a cluster of pixels (above a threshold of, say, 250, to filter out noise resulting from the initial flagging stage) at t to an overlapping cluster at t+1.

So if i just deal with the first cluster at t first of all, using the histogram tip i picked up from you...(fld1 and fld2 are two successive timesteps of flag data from my much larger parent dataset (intarr(1133,751)))

```
;find blobs at t
blobs1=label_region(fld1)
pop1=histogram(blobs1)
x1=where(pop1 ge 250)
a=where(blobs1=x1[1])

;select first cluster only, ignore the rest
cluster1=intarr(1133,751)
cluster1(a)=1

;find blobs at t+1
blobs2=label_region(fld2)

;-set all blobs that pass size threshold equal to 1 so that overlap
can be found with cluster1
pop2=histogram(blobs2)
x2=where(pop2 ge 250)

tmp=intarr(1133,751)

for i=0,n_elements(x2)-2 do begin
    j=where(blobs2 eq x2[i+1])
    tmp(j)=1
endfor

;add cluster1 to tmp to work out overlap (overlapping regions have
value 2)
overlap=cluster1+tmp
```

;assign cluster at t+1 to which this overlapping region belongs the same identifier as the cluster at t

errr, here i get stuck!

Ian