Subject: Re: Given many images, find bounding box
Posted by Dick Jackson on Fri, 05 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Herbert wrote:

> I have many spatially seperated (but maybe overlapped) images patches
> which I would like to make a big mosaic image with them.
>
> For each of the patch, I have the coor. (lat, lon) of the upper left
> hand corner and the # of rows and # of cols. I would like to find the
> biggerest bounding box that will contain all the patches.
>
> Can anyone point me to any existing algorithm that will find this
> bounding box

This is pretty straightforward, but I have to make a few assumptions:

- you have four arrays (lat, lon, rows, cols) containing values
  for each of your patches
- all four are measured in pixels
- 'lat' increases as you go up, 'lon' increases as you go to the right
  (no wrapping at lat +/- 90 or lon +/- 180 here, do you need that?
   My, that would be interesting...)
- you want the *smallest* bounding box that contains the patches

Then the four edges of that bounding box are:

left = Min(lon)
top = Max(lat)
right = Max(lon + cols - 1)
bottom = Min(lat - rows + 1)

Hope this helps!

Cheers,
-Dick

Dick Jackson Fanning Software Consulting, Canadian Office
djackson@dfanning.com Calgary, Alberta Voice/Fax: (403) 242-7398
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Given many images, find bounding box
Posted by hhpt on Fri, 05 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Hi Dick,

Well, this is not correct, for example, if I havbe only one patch,
the upper left hand corner has the coordinate (1,10) (lon, lat) and the
box has lon_size = 6 and lat_size = 8

then the expected answer should be:
  left = 1
  right 7
  top = 10
  bottom = 2
but using your formular the answers would be:
  left = 1
  right 6
  top = 10
  bottom = 3

Also, it will be nice some how if it could take care of the +/- degrees
too..... seems tricky....

-- Herbert

> This is pretty straightforward, but I have to make a few assumptions:
>
> - you have four arrays (lat, lon, rows, cols) containing values
>   for each of your patches
> - all four are measured in pixels
> - 'lat' increases as you go up, 'lon' increases as you go to the right
>   (no wrapping at lat +/- 90 or lon +/- 180 here, do you need that?
>   My, that would be interesting...)
> - you want the *smallest* bounding box that contains the patches
>
> Then the four edges of that bounding box are:
>
> left = Min(lon)
> top = Max(lat)
> right = Max(lon + cols - 1)
> bottom = Min(lat - rows + 1)
>
> Hope this helps!
>
> Cheers,
> -Dick
>
> Dick Jackson Fanning Software Consulting, Canadian Office
> djackson@dfanning.com Calgary, Alberta Voice/Fax: (403) 242-7398
>   Coyote's Guide to IDL Programming: http://www.dfanning.com/
>
>

Subject: Re: Given many images, find bounding box
Posted by Dick Jackson on Sat, 06 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

>> This is pretty straightforward, but I have to make a few assumptions:
>>
>> - you have four arrays (lat, lon, rows, cols) containing values
>> for each of your patches
>> - all four are measured in pixels
>> - 'lat' increases as you go up, 'lon' increases as you go to the right
>> (no wrapping at lat +/- 90 or lon +/- 180 here, do you need that?
>> My, that would be interesting...)
>> - you want the *smallest* bounding box that contains the patches
>>
>> Then the four edges of that bounding box are:
>>
>> left = Min(lon)
>> top = Max(lat)
>> right = Max(lon + cols - 1)
>> bottom = Min(lat - rows + 1)

Herbert wrote:
> Hi Dick,
>
> Well, this is not correct, for example, if I havbe only one patch,
> the upper left hand corner has the coordinate (1,10) (lon, lat) and the
> box has lon_size = 6 and lat_size = 8
>
> then the expected answer should be:
> left = 1
> right 7
> top = 10
> bottom = 2
> but using your formular the answers would be:
> left = 1
> right 6
> top = 10
> bottom = 3

Clearly, if that's the answer you need, just remove the "+ 1" and "- 1" from the calculations. I notice that your earlier use of 'cols' and 'rows' has changed to lon_size and lat_size. If you are calculating precise

measurements of lat/lon, then of course these would be formulas to use:

left = Min(lon)
top = Max(lat)
right = Max(lon + lon_size)
bottom = Min(lat - lat_size)

From your original description using 'rows' and 'cols', I was working with another assumption, that all these numbers refer to pixel locations and sizes, and you would use the bounds in order to, say, draw a box that includes all of them. In your example, I took it to mean the pixel at [1,10] is at top-left of the patch, and the patch is 6x8 pixels, thus extending to (counting fingers...) [6,3], *inclusive*. If you use my first calculations, the lines will coincide with the outermost pixels all around. The new calculations would give a box that coincides at top and left, but extends one pixel beyond at right and bottom.

This distinction of pure-measurements vs. pixel-counting is surely one of the greatest sources of off-by-one errors in all computing!

> Also, it will be nice some how if it could take care of the +/- degrees
> too..... seems tricky....

I believe the first tricky part is to precisely define the question.

--
Cheers,
-Dick

Dick Jackson Fanning Software Consulting, Canadian Office
djackson@dfanning.com Calgary, Alberta Voice/Fax: (403) 242-7398
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Given many images, find bounding box
Posted by Dick Jackson on Mon, 08 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

[it seems that my first try at sending this message didn't get out, reposting:]

>> This is pretty straightforward, but I have to make a few assumptions:
>>
>> - you have four arrays (lat, lon, rows, cols) containing values
>> for each of your patches
>> - all four are measured in pixels
>> - 'lat' increases as you go up, 'lon' increases as you go to the right
>> (no wrapping at lat +/- 90 or lon +/- 180 here, do you need that?

>> My, that would be interesting...)
>> - you want the *smallest* bounding box that contains the patches
>>
>> Then the four edges of that bounding box are:
>>
>> left = Min(lon)
>> top = Max(lat)
>> right = Max(lon + cols - 1)
>> bottom = Min(lat - rows + 1)

Herbert wrote:
> Hi Dick,
>
> Well, this is not correct, for example, if I havbe only one patch,
> the upper left hand corner has the coordinate (1,10) (lon, lat) and the
> box has lon_size = 6 and lat_size = 8
>
> then the expected answer should be:
>   left = 1
>   right 7
>   top = 10
>   bottom = 2
> but using your formular the answers would be:
>   left = 1
>   right 6
>   top = 10
>   bottom = 3

Clearly, if that's the answer you need, just remove the "+ 1" and "- 1" from the calculations. I notice that your earlier use of 'cols' and 'rows' has changed to lon_size and lat_size. If you are calculating precise measurements of lat/lon, then of course these would be formulas to use:

left = Min(lon)
top = Max(lat)
right = Max(lon + lon_size)
bottom = Min(lat - lat_size)

From your original description using 'rows' and 'cols', I was working with another assumption, that all these numbers refer to pixel locations and sizes, and you would use the bounds in order to, say, draw a box that includes all of them. In your example, I took it to mean the pixel at [1,10] is at top-left of the patch, and the patch is 6x8 pixels, thus extending to (counting fingers...) [6,3], *inclusive*. If you use my first calculations, the lines will coincide with the outermost pixels all around. The new calculations would give a box that coincides at top and left, but extends one pixel beyond at right and bottom.

This distinction of pure-measurements vs. pixel-counting is surely one of the greatest sources of off-by-one errors in all computing!

> Also, it will be nice some how if it could take care of the +/- degrees
> too..... seems tricky....

I believe the first tricky part is to precisely define the question.

--
Cheers,
-Dick

Dick Jackson Fanning Software Consulting, Canadian Office
djackson@dfanning.com Calgary, Alberta Voice/Fax: (403) 242-7398
Coyote's Guide to IDL Programming: http://www.dfanning.com/
