Subject: Algorithm for lat/lon searching
Posted by Paul Van Delst[1] on Fri, 18 Aug 2006 14:50:56 GMT
View Forum Message <> Reply to Message

Hello,

I want to implement a global *land* surface emissivity database (as a LUT) into a radiative transfer code. Fir simplicity the database is simply gridded by lat/lon (land and sea). Due to memory limitations, I want to only keep the land gridboxes in my lookup table. Obviously, doing this complicates searching for the actual lat/lon element since they're no longer stored on a grid.

What I'm looking for is a simple and/or quick method for searching a somewhat irregularly spaced database for particular points. In the IDL newsgroup there was recently a discussion above finding unique number pairs (lat->"high" portion of 64 bit int, lon->"low" portion) and I was thinking that would provide a searchable database. By converting the lat/lon pair to a unique number, e.g.

JD Smith wrote:
<IDL code follows>
> epsilon=1.e-7 ; difference in degrees for equality
> lat_lon = ulong64((lat+90.)/epsilon) + ishft(ulong64(lon/epsilon),32)

the resultant lat_lon array being simple to search.

An additional problem is that, since this data will be used for satellite data assimilation and satellites tend to scan "diagonally" across lat/lon, adjacent/close-by *geographical* grid elements will be accessed and it's not clear to me that the above lat/lon orgainisation will put elements separated by a short physical distance anywhere near each other in the lat_lon array.

I will be playing with and testing this over the coming days, but I wanted to pick the brains of folks out there in advance.

Thanks for any suggestions/advice,

cheers,

paulv

p.s. Since the final code needs to be Fortran95, I set followups to comp.lang.fortran

--
Paul van Delst Ride lots.
CIMSS @ NOAA/NCEP/EMC Eddy Merckx
Ph: (301)763-8000 x7748
Fax:(301)763-8545

Subject: Re: Algorithm for lat/lon searching
Posted by K. Bowman on Fri, 18 Aug 2006 21:35:56 GMT
View Forum Message <> Reply to Message

In article <ec5b4m$rf1$1@news.nems.noaa.gov>,
 Paul van Delst <Paul.vanDelst@noaa.gov> wrote:

> Hello,
>
> JD Smith wrote:
>> On Fri, 18 Aug 2006 10:50:56 -0400, Paul van Delst wrote:
>>
>>> Hello,
>>>
>>> I want to implement a global *land* surface emissivity database (as a LUT)
>>> into a radiative transfer code. Fir simplicity the database is simply
>>> gridded by lat/lon (land and sea). Due to memory limitations, I want to
>>> only keep the land gridboxes in my lookup table. Obviously, doing this
>>> complicates searching for the actual lat/lon element since they're no
>>> longer stored on a grid.

You can do something rather like reverse indices in HISTOGRAM. Treat your data as a 1-D array, and make an index that lets you access the relevant lines of data. This is similar to run-length encoding to compress an image. (A land-sea mask is a kind of binary image.) You'll trade-off searching the index for storage. In these days of cheap memory, is that a good trade-off?

Cheers, Ken

Subject: Re: Algorithm for lat/lon searching
Posted by Terence on Fri, 18 Aug 2006 22:28:17 GMT
View Forum Message <> Reply to Message

Can I suggest building two tables of the globe, which maps surface area units as a small number of very approximately square areas (except near the poles when they could become triangular)? The two tables refer to two sets of such maps that overlap so that the centre of one area is on the boundary of a similar area of the second map.
The northern and southern triangles are just displaced by longitude in a similar relationship.

You build your two tables of the globe as area numbers which corresponding to a numbering of the centre points based on known latitude and longitude. Now you take your table of lat/lon points data and assign the two individual map area numbers to each point on one pass by rounding and truncating the exact coordinates so that the high-order digits match the area centre lat/lon numbers and allow

direct assignment of the areas they have in common.

On a second pass, all numbers with the same index in either mapping are close neighbours. If both indices are identical they are very close neighbours within half an area unit. You choose a mapping resolution suitable for the work to be done

Then you could order the data points by the new index numbers and so process the globe data in a sequential manner that treats neighbouring points in order.

Terence Wright

Subject: Re: Algorithm for lat/lon searching
Posted by Bob Walton on Sun, 20 Aug 2006 03:18:00 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:

> Hello,
>
> JD Smith wrote:
>
>> On Fri, 18 Aug 2006 10:50:56 -0400, Paul van Delst wrote:
>>

...

> The spatial index searching that Gordon Sande mentioned might also be
> the go (if i can figure out how to do it).

Don't forget to check CPAN. Maybe something like the Algorithm::QuadTree module would help.

...

> paulv
>
>

--
Bob Walton
Email: http://bwalton.com/cgi-bin/emailbob.pl
