
Subject: Re: Algorithm for lat/lon searching
Posted by [Paul Van Delst\[1\]](#) on Fri, 18 Aug 2006 21:23:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

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. For 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.

>

> Here's a simple notion:

>

> Why not develop a "whole earth grid" in whatever binning and projection is

> useful (an equal area projection comes to mind), run all your land points

> (only) through HIST_ND, store the resulting REVERSE_INDICES, and then, for

> a given lat/lon, look up its position in the multi-dimensional reverse

> index vector, and read out the emissivity data points. You don't say how

> much information each of those emissivity data points would include, but

> storing a reverse index vector is linear in the number of bins, and would

> be much faster to access than sorting through constantly.

>

>> p.s. Since the final code needs to be Fortran95, I set followups to

>> comp.lang.fortran

>

> Oh, well, that's a different story then, since you'd have to write

> your own HISTOGRAM from scratch ;).

Well, the final code that accesses the LUT needs to be f95, but I can prepare the datafile offline however I like. If I only have to histogram the data once, and store those indices in the datafile along with the data itself, that's fine.

If I understand what you're saying, the problem is that I can't have a "whole earth grid" - I only want land data points. I.e. I might have an array of 720x360 (lon x lat) that I would apply a land/sea mask to. That would give me, say, 30% of the original data where there is no longer any regular grid (except within continental areas I guess).

> I guess it depends on your

> desired bin size, then. If you can bin the whole earth (or whatever

> portion thereof you're discussing) into say, 2¹⁶ points, then map a

> given LAT/LON (or other more convenient re-projected coordinate pair)

- > to two 8bit integers (aka 1 16bit int, similar to what I showed for
- > 64bit integers), you could keep a simple 65536 element hash table,
- > each element of which would point to the data contained (using
- > whatever F95 magic may or may not exist to do that: pointers to a
- > linked lists would come to mind for us C programmers). You could
- > implement more than 8bit of gridding per dimension, at the cost of
- > memory. 10bits each (1024 elements) would only occupy about 4MB.

My initial thoughts were along that line. Put together a list of land position keys based on your packing the lat/lon into a single number, use that algorithm to compute the required key, and then search the hash table for the emissivity data values.

Whenever I need to do any searching I always try to get a second (and third, ...) opinion. :o)

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

cheers,

paulv

--

Paul van Delst Ride lots.

CIMSS @ NOAA/NCEP/EMC

Eddy Merckx

Ph: (301)763-8000 x7748

Fax:(301)763-8545
