Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by Jim Pendleton on Fri, 03 May 2013 02:39:10 GMT
View Forum Message <> Reply to Message

On Monday, April 29, 2013 5:26:44 PM UTC-6, Paul van Delst wrote:
> Hello,
>
>
>
> The subject line initially read "Function graphics equivalent of PLOTS?"
>
> but I changed it to what I really want to do.
>
>
>
> I have an older direct graphics procedure that plots individual data
>
> points (satellite data) on a map, where the colour of each distinct
>
> field-of-view (FOV) is a function of the measured quantity (say,
>
> radiance or temperature).
>
>
>
> This is achieved by creating the global map, then looping over each
>
> observation and plotting it on the map via PLOTS setting the colour
>
> separately as needed for each plot. Takes about 0.5 seconds to display a
>
> couple of orbits of data.
>
>
>
> Standard sort of stuff IDL is used for, right?
>
>
>
> For grins I thought I'd alter the code to do it using function graphics.
>
> But, how does one do that? There's no equivalent of PLOTS. And besides,
>
> plotting one point at a time in function graphics (when you have more
>
> than a couple hundred points) takes forever (15minutes and counting
>
> right now, for pete's sake).

>
>
>
> To reiterate my question: How would one plot satellite tracks of
>
> individual FOV data on a global map? E.g. a single orbit of polar
>
> orbiter data?
>
>
>
> It used to be a trivial thing to do in direct graphics. And the IDL help
>
> is useless unless you want to register a nice regular image with a map
>
> projection.
>
>
>
> cheers,
>
>
>
> paulv
>
>
>
> p.s. I'm still at IDL v8.2 and I'm getting really really tired of
>
> waiting many minutes for plots to display (that take fractions of a
>
> second in DG). I'm hoping the latest versions of IDL have sped up
>
> function graphics display by at least several orders of magnitude. Is
>
> that the case?

With respect to Chris' example, the data is greatly oversampled relative to the display pixels available in the window so large blobs are not really appropriate. Try simply changing the IDLgrSymbol to create a single point via "oSym = IDLgrSymbol(3)", which is more like DG PLOTS with PSYM = 3. That'll reduce both the execution time and memory use significantly.