Subject: Re: Can this be done using CALL_FUNCTION?
Posted by Struan Gray on Wed, 08 Mar 2000 08:00:00 GMT
View Forum Message <> Reply to Message

J.D. Smith, jdsmith@astro.cornell.edu writes:

[good advice deleted]

There are some tricks you can play to speed things up for specific filtering operations.

If you do ever want to filter the 'vertical' dimension, unroll the 2D dimensions and loop once over a single index rather than use two nested loops. This is dramatically faster for any multispectral data large enough to be called an image.

Histograms are fast in IDL, loops are slow. One trick is to multiply each slice of your data by a large enough number that data in the slices does not overlap. Doing a single histogram on the whole array then gives individual histograms dotted along the bin axis - extracting them is easy. The efficiency of doing this depends on how much memory you have and how many spectral slices need to be looped over. If your data is already maximum precision this won't help, but it rarely is.

For filters which can be expressed as convolution with a kernal, use a 3D kernal with non-zero values only in the central plane. Again, this seem daft, but loops are so inefficient in IDL that for a reasonable number of bands this is faster than 2D kernals and a loop.

As JD pointed out, an object representation of your data makes using and customising this sort of thing much easier than writing generic routines that loop through arbitrary data arrays. A data object *knows* what the 2D slices represent, and which filtering operations are appropriate for which slices; a generic looping routine doesn't. Repetitive typing can be avoided by appropriate use of object inheritance.

Struan