Subject: Re: Can this be done using CALL_FUNCTION?
Posted by John-David T. Smith on Tue, 07 Mar 2000 08:00:00 GMT
View Forum Message <> Reply to Message

edward.s.meinel@aero.org wrote:
>
> OK, here's one for the IDL gurus (since I'm only a little guru would
> that make me a gu-gu?)...
>
> I am working with spectral images. Unfortunately, IDL is geared toward
> multidimensional data in which all of the dimensions are the same type
> (i.e. spatial, spectral, frequency...) but it doesn't like to operate on
> data with mixed dimensions, such as a multispectral image (unless I'm
> missing something really obvious).
>

What is it about multi-spectral data that IDL doesn't like? It certainly
doesn't care about whether a single dimension of your data array represents
spatial, temporal, or spectral changes... maybe I'm missing something. The IDL
data types are tied to machine types: longs, shorts, floats, doubles, etc...
mixing these types in a single array *is* impossible.

E.g., a 4-d array of floats in which axes 1 & 2 are spatial (e.g. latitide and
longitude), axis 3 is spectral (say 10 different bands), and axis 4 is temporal
(100 days worth of data at once per day) is perfectly acceptable. It's up to
you to keep track of which dimensions are which, but I don't see the problem...
It is true that certain IDL routines operate on images, or require other
specially formatted or dimensioned data, but how is it to know which dimensions
you are interested in without specifically telling it?

Maybe you could give us an example in which this kind of generality would be
useful.

JD

--
```
 J.D. Smith                        |*|     WORK: (607) 255-5842
 Cornell University Dept. of Astronomy |*|          (607) 255-6263
 304 Space Sciences Bldg.              |*|      FAX: (607) 255-5875
 Ithaca, NY 14853                  |*|
```

Subject: Re: Can this be done using CALL_FUNCTION?
Posted by Martin Schultz on Tue, 07 Mar 2000 08:00:00 GMT
View Forum Message <> Reply to Message

edward.s.meinel@aero.org wrote:
>

> OK, here's one for the IDL gurus (since I'm only a little guru would
> that make me a gu-gu?)...
>
> I am working with spectral images. Unfortunately, IDL is geared toward
> multidimensional data in which all of the dimensions are the same type
> (i.e. spatial, spectral, frequency...) but it doesn't like to operate on
> data with mixed dimensions, such as a multispectral image (unless I'm
> missing something really obvious).
>
> Rather than having to rewrite every bloody function/procedure for
> spectral imagery, I was hoping it would be possible to write a generic
> wrapper function along the lines of:
>
> FUNCTION spatial_function, name, image, other_stuff
> im_size = SIZE(image)
> CASE im_size OF
> [... rest of code snipped][/color]

Ed,

in a perfect world you could do this, although you should add
_EXTRA=e to your wrapper to allow for keywords. Unfortunately,
several of the built-in (or library) functions are not forgiving
when it comes to the wrong number of parameters and keywords. Therefore,
you will probably not get around doing something (ugly) like:

```
function spatial_function, name, image, arg1, arg2, arg3, arg4, _EXTRA=e

   im_size = SIZE(image)
   CASE im_size OF
     ...
     CASE n_params() OF
      2 : CALL_FUNCTION(name,image)
      3 : CALL_FUNCTION(name,image,arg1)
      4 : CALL_FUNCTION(name,image,arg1,arg2)
      ...
     ENDCASE
   ...
```

A way around this? Well, you could rewrite all the functions you would
potentially call from such a wrapper and give them extra arguments which
are simply ignored if they are not needed.
Example:
change
FUNCTION Correlate, X, Y, Covariance = Covariance, Double = Double
into
FUNCTION Correlate, X, Y, DUM1, DUM2, DUM3, DUM4, DUM5, DUM6, $

Covariance = Covariance, Double = Double

And if the function doesn't have keywords, add _EXTRA=e to it.

Good luck,
Martin

```
--
 [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[ [[[[[[[
[[ Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie    [[
[[                      Bundesstr. 55, 20146 Hamburg             [[
[[                      phone: +49 40 41173-308                  [[
[[                      fax:   +49 40 41173-298                  [[
[[ martin.schultz@dkrz.de                                        [[
 [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[ [[[[[[[
```

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

Struan Gray (struan.gray@sljus.lu.se) and J.D. Smith,
(jdsmith@astro.cornell.edu) write their usual excellent advice:

> [good advice deleted]

You guys are probably scaring poor Ed to death with
all this talk about objects. But it is very good
advice, indeed. I'll bet the ENVI guys are wishing
they had objects to work with when they first started
writing ENVI. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

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

Subject: Re: Can this be done using CALL_FUNCTION?
Posted by John-David T. Smith on Wed, 08 Mar 2000 08:00:00 GMT

View Forum Message <> Reply to Message

edward.s.meinel@aero.org wrote:
>
> In article <38C53FCA.A89BD43E@astro.cornell.edu>,
> "J.D. Smith" <jdsmith@astro.cornell.edu> wrote:
>> edward.s.meinel@aero.org wrote:
>>>

>>> I am working with spectral images. Unfortunately, IDL is geared toward
>>> multidimensional data in which all of the dimensions are the same type
>>> (i.e. spatial, spectral, frequency...) but it doesn't like to operate
>>> on data with mixed dimensions, such as a multispectral image.
>>
>> What is it about multi-spectral data that IDL doesn't like?
>
> You answered your own question...
>
>> It's up to
>> you to keep track of which dimensions are which,
>
> This gets to be a real pain after the umpteenth time of checking the
> number of dimensions and setting up the different cases of TRUE.
>
>> It certainly
>> doesn't care about whether a single dimension of your data array
>> represents spatial, temporal, or spectral changes...
>> maybe I'm missing something.
>
> No, IDL doesn't care about these issues when _creating_ the array, but
> it certainly does care when _operating_ on the array.
>
>> It is true that certain IDL routines operate on images, or require
>> other specially formatted or dimensioned data, but how is it to know
>> which dimensions you are interested in without specifically telling it?
>
> Well, I was thinking about something along the lines of the example I
> gave.
>
>> Maybe you could give us an example in which this kind of generality
>> would be useful.
>
> I did. How about HISTOGRAM? FFT? Median filter? ...
>

I think IDL can do this for you, with a very reasonable amount of work. Your
basic reasoning goes: I don't like to remember which dimensions of my array
correspond to which types -- spatial, temporal, or frequency, etc. Obviously,
you don't want to take an FFT of a slice of a multi-spectral hypercube which has
time as one axis and space as another... but IDL doesn't care if you pass it
this data. It's happy to take a time-space FFT or median, or histogram, as long
as the calling conventions and input machine-types/dimensions are honored. That
might not produce any sensible results, but IDL certainly is happy to *operate*
on, and not just *create* arrays of any abstract type, since it knows nothing of
those "types". In your spatial_function() routine, you try to make the type
ordering in your data general. As with most problems like this, the solution is
in the data representation phase, not the operation phase. Simply adopt a

convention such as:

first two dimensions: spatial
next: time
next: frequency

and manipulate all input data to conform to this convention (with reform, for example). This might be more natural in an object framework, which hides the details, and lets you make custom mapping methods... an example session might look like:

```
a=obj_new("thisdata.dat",ORDERING=1) ; read in data with a specific dimensional
ordering
tvscl, a->fft(/SPACESPACE,PLANE=4)   ; return the FFT of the 4th Space-space
slice.
cube=a->median(/SPACE1TIME)          ; return the median cube: 1st spatial x
time
```

etc.

If you have header keywords which give the dimensional ordering info, the ORDERING could be omitted. You could also obviously preserve that native orientation of your data, and use that information (recorded in the object's class data) to do something more along the lines of your "spatial_function" routine, but I prefer as simple and standardized a data product as possible, to lighten the programming load.

Good luck,

JD

--
 J.D. Smith |*| WORK: (607) 255-5842
 Cornell University Dept. of Astronomy |*| (607) 255-6263
 304 Space Sciences Bldg. |*| FAX: (607) 255-5875
 Ithaca, NY 14853 |*|

Subject: Re: Can this be done using CALL_FUNCTION?
Posted by edward.s.meinel on Wed, 08 Mar 2000 08:00:00 GMT

View Forum Message <> Reply to Message

In article <38C53FCA.A89BD43E@astro.cornell.edu>,
  "J.D. Smith" <jdsmith@astro.cornell.edu> wrote:
> edward.s.meinel@aero.org wrote:
>>

>> I am working with spectral images. Unfortunately, IDL is geared toward
>> multidimensional data in which all of the dimensions are the same type
>> (i.e. spatial, spectral, frequency...) but it doesn't like to operate
>> on data with mixed dimensions, such as a multispectral image.
>
> What is it about multi-spectral data that IDL doesn't like?

You answered your own question...

> It's up to
> you to keep track of which dimensions are which,

This gets to be a real pain after the umpteenth time of checking the
number of dimensions and setting up the different cases of TRUE.

> It certainly
> doesn't care about whether a single dimension of your data array
> represents spatial, temporal, or spectral changes...
> maybe I'm missing something.

No, IDL doesn't care about these issues when _creating_ the array, but
it certainly does care when _operating_ on the array.

> It is true that certain IDL routines operate on images, or require
> other specially formatted or dimensioned data, but how is it to know
> which dimensions you are interested in without specifically telling it?

Well, I was thinking about something along the lines of the example I
gave.

> Maybe you could give us an example in which this kind of generality
> would be useful.

I did. How about HISTOGRAM? FFT? Median filter? ...

Ed Meinel

Sent via Deja.com http://www.deja.com/
Before you buy.

Subject: Re: Can this be done using CALL_FUNCTION?
Posted by davidf on Thu, 09 Mar 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Struan Gray (struan.gray@sljus.lu.se) writes:

> Of course, you should think hard about whether you have the
> maturity to cope with the screaming crowds of nubile IDL object
> groupies...

Damn, Struan. I was saving this little pearl of wisdom
for the book. :-(

Cheers,

David

P.S. Let's just say this is NOT amoung the reasons I
use when my wife asks me why I've just spent ANOTHER
evening camped in front of the computer.

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Can this be done using CALL_FUNCTION?
Posted by Struan Gray on Thu, 09 Mar 2000 08:00:00 GMT

View Forum Message <> Reply to Message

edward.s.meinel@aero.org writes:

> How many of you started using the HANDLE_* routines
> as soon as IDL4 came out?

Guilty as charged. Fanning? Who he? :-)

When I first started programming in IDL I was porting a large
microscope control and analysis program from Unix IDL to run on cheap
machines like Macs and PCs. Working my way through that spaghetti
gave me a deep and abiding hatred of common blocks. 4.0, and the
ability to hang dynamic data structures off a single system variable
was like manna from heaven.

The only problem with objects is that the possibilites are endless,
and it's easy to get lost footling around with high-level abstract
objects when what you really need is something to draw a single red
dot *today*. Objects are certainly here to stay, and even if your
users see no benefits you as a programmer will find your life made
much easier once you start to use them. Your looping procedure is an
almost exact copy of the classic situation used in most textbooks to

show the advantages of Object-orientation.

Of course, you should think hard about whether you have the maturity to cope with the screaming crowds of nubile IDL object groupies...

Struan

Subject: Re: Can this be done using CALL_FUNCTION?
Posted by davidf on Thu, 09 Mar 2000 08:00:00 GMT

View Forum Message <> Reply to Message

Ed Meinel (edward.s.meinel@aero.org) writes:

> No, I'm not too scared... I usually like to wait one major upgrade
> before starting to use a new capability. This gives RSI a chance to fix
> all the bugs that David found ;-)

Well, that's probably good advice. The only trouble with it is that
if everyone follows it no bugs are found or fixed. :-(

> How many of you started using the HANDLE_* routines as soon as
> IDL4 came out?

It is funny how these things evolve. I don't think handles
would have happened at all, except that I started to advertise
that I could teach people how to use "pointers" in IDL classes.
We were using the user value of unrealized base widgets as
pointers or handles. I remember being surrounded by a bunch
of developers in the hallway once. They had one of my brochures
and they were in an ugly mood. "What the hell is this pointer
nonsense, Fanning?", they demanded.

"Uh, well, they aren't *really* pointers, but look what we
can do with them", I said. Next thing I knew we had an
unrealized base widget with everything stripped off it
except the user value and renamed a "handle". And, of course,
since that really didn't do *exactly* what we wanted either,
we soon had real pointers. (Well, as real as you are likely
to get in a language like IDL.)

But I do like the direction the language is evolving in.
I just wish they wouldn't make such a big mystery of all
the new object graphics capabilities. I just can't figure
why they want to hide this light under their bushel basket.
I'd be telling the whole world about it. Especially those

young pups who think Matlab is the cat's meow.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Can this be done using CALL_FUNCTION?
Posted by edward.s.meinel on Thu, 09 Mar 2000 08:00:00 GMT
View Forum Message <> Reply to Message

davidf@dfanning.com (David Fanning) wrote:
> Struan Gray (struan.gray@sljus.lu.se) and J.D. Smith,
> (jdsmith@astro.cornell.edu) write their usual excellent advice:
>
>> [good advice deleted]

Well, actually it is

    [good advice not repeated]

You didn't need to go and _delete_ the good advice! I might want to use it...

>
> You guys are probably scaring poor Ed to death with
> all this talk about objects. But it is very good
> advice, indeed. I'll bet the ENVI guys are wishing
> *they* had objects to work with when they first started
> writing ENVI. :-)
>

No, I'm not too scared... I usually like to wait one major upgrade
before starting to use a new capability. This gives RSI a chance to fix
all the bugs that David found ;-) and decide whether or not to keep it
in IDL. How many of you started using the HANDLE_* routines as soon as
IDL4 came out?

Ed 'biting the bullet' Meinel