
Subject: Can this be done using CALL_FUNCTION?
Posted by [edward.s.meinel](#) on Mon, 06 Mar 2000 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

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[0] OF
  1: BEGIN      ; vector
    data = CALL_FUNCTION(name, image, other_stuff)
    RETURN, data
  END

  2: BEGIN      ; image
    data = CALL_FUNCTION(name, image, other_stuff)
    RETURN, data
  END

  3: BEGIN      ; spectral image
    CASE true OF
      1: BEGIN      ; BIP
        bands = im_size[1]
        pixels = im_size[2]
        lines = im_size[3]
        data = image
        FOR i = 0, bands - 1 DO BEGIN
          dummy = REFORM(image[i, *, *])
          dummy = CALL_FUNCTION(name, dummy, other_stuff)
          data[i, *, *] = dummy
        ENDFOR
      END
      2: BEGIN      ; BIL
        bands = im_size[2]
        pixels = im_size[1]
        lines = im_size[3]
```

```

data = image
FOR i = 0, bands - 1 DO BEGIN
    dummy = REFORM(image[* , i, *])
    dummy = CALL_FUNCTION(name, dummy, other_stuff)
    data[* , i, *] = dummy
ENDFOR
END

ELSE: BEGIN          ; BSQ
    bands = im_size[3]
    pixels = im_size[1]
    lines = im_size[2]
    data = image
    FOR i = 0, bands - 1 DO BEGIN
        dummy = REFORM(image[* , *, i])
        dummy = CALL_FUNCTION(name, dummy, other_stuff)
        data[* , *, i] = dummy
    ENDFOR
END
ENDCASE
RETURN, data
END

ELSE: BEGIN          ; error
    data = DIALOG_MESSAGE('The input array is not an image.')
    RETURN, 0
END

```

END

Then I could do things like:

```

hist = SPATIAL_FUNCTION('histogram', image)
mw = SPATIAL_FUNCTION('median', image, 5)

```

Is there any hope for me???

Ed Meinel

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