
Subject: Re: IDL Indexing 2D->3D

Posted by [Michael Galloy](#) on Wed, 26 Mar 2014 22:48:44 GMT

[View Forum Message](#) <> [Reply to Message](#)

On 3/26/14, 3:55 PM, forthewynn@gmail.com wrote:

> Hi All,

>

> I have a peculiar predicament that I am trying to solve...unfortunately, the nature of the problem does not allow me to google it easy, and IDL is not my main programming language.

>

> Let's say I have a 3x3x15 array. I use the where command to find certain values in the first 'frame' (third dimension)--note, this is the main stickler, in that I can only search for these values in the first 'frame' as they are not present throughout the whole array.

>

> So, the where command spits out three points in example: [2,3,5].

>

> Using array_indices with the dimension of the 3x3x15 array, I get the locations of these points, like [[0,2,0],[1,0,0],[1,2,0]]. This is all fine and dandy so far.

>

> My problem is that I need to translate these indices to the corresponding points in all of the 'frames'. In other words, I need [[0,2,0],[1,0,0],[1,2,0],[0,2,1],[1,0,1],[1,2,1],[0,2,2],[1,0,2],[1,2,2],...through all 15 levels of the third dimension].

>

> I have found that I can make a loop with the index being the third dimension to accomplish this, or alternatively I can make an array composed of [where command output, where command output + dimension_1*dimension_2*1, where command output + dimension_1*dimension_2*2, etc.]. But the time to complete these increases heavily as the dimensions of my original array increase (800x800x400 in the case of my project).

>

> Thus, I am wondering if anyone might have a shortcut that can be accomplished in one statement (since IDL is usually very good with matrices) like theoretically:

>

> ORIGINAL_ARRAY=FINDGEN(800,800,400) ; EXAMPLE

> TEMP=WHERE(ORIGINAL_ARRAY EQ VALUE-ONLY-IN-FIRST-FRAME,CT)

> TEMP_IND=ARRAY_INDICES(ORIGINAL_ARRAY,TEMP)

>

> DATA_ABOVE_WHERE_POINTS=ORIGINAL_ARRAY[[TEMP_IND[0,*],TEMP_IND[1,*]],*]

>

>

> that yields HELP,DATA_ABOVE_WHERE_POINTS: DOUBLE = ARRAY[CT,400].

>

>

> Obviously the above does not work because of the way IDL indexes (this yields a [CNT,CNT] array). Thanks for viewing and I appreciate any pointers.

>

Is this what you want?

```
; set an array, 4rd dimension is number of frames
a = findgen(3, 3, 15)

dims = size(a, /dimensions)

; some condition that only holds in the first frame, could also
; index a to make sure indices fall in first frame
ind = where(a gt 1.0 and a lt 5.0, count)

; make a copy of ind for each frame and copy of the offset for each
; count, add them up
single_frame_ind = rebin(reform(ind, count, 1L), count, dims[2])
frame_offsets = rebin(reform(lindgen(dims[2]) * dims[0] * dims[1], $
                           1L, dims[2]), $
                      count, dims[2])
all_ind = reform(single_frame_ind + frame_offsets, count * dims[2])

; convert back to coordinates instead of indices
coordinates = array_indices(a, all_ind)

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician
Tech-X Corporation
```
