Subject: IDL Indexing 2D->3D
Posted by forthewynn on Wed, 26 Mar 2014 21:55:41 GMT
View Forum Message <> Reply to Message

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 programing 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_I ND[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.

Subject: Re: IDL Indexing 2D->3D
Posted by Michael Galloy on Wed, 26 Mar 2014 22:48:44 GMT

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 programing 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_I ND[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

Subject: Re: IDL Indexing 2D->3D
Posted by cgguido on Thu, 27 Mar 2014 15:53:59 GMT
View Forum Message <> Reply to Message

On Wednesday, March 26, 2014 4:55:41 PM UTC-5, forth...@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 programing 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_I ND[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.

I find this simpler, provided you have enough memory to make a copy of your initial array:

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

;repeat first slice N_z times:
dim=size(a,/dim)
a0=a[*,*,0]
arepeat=rebin(a0, dim)

;Search for values:
w = where(arepeat gt 1.0 and arepeat lt 5.0, count)

:Get coordinates:
xyz = array_indices(a, w)


;I've checked and xyz is identical to Mike's coordinates. For small xy dims the above is faster, for medium and large xy dims Mike's faster.
```

Subject: Re: IDL Indexing 2D->3D
Posted by forthewynn on Fri, 28 Mar 2014 15:04:57 GMT
View Forum Message <> Reply to Message

Thanks everyone for the help. Both methods do work, but I will go with Mike's as my dimensions are quite large. Best wishes!

Subject: Re: IDL Indexing 2D->3D
Posted by cgguido on Sat, 29 Mar 2014 01:53:25 GMT
View Forum Message <> Reply to Message

On Friday, March 28, 2014 10:04:57 AM UTC-5, forth...@gmail.com wrote:
> Thanks everyone for the help. Both methods do work, but I will go with Mike's as my dimensions are quite large. Best wishes!

Best of luck!

When in doubt, always go with Mike or David or JD Smith or Lajos..., or Coyote if you're that lucky. When not in doubt, also go with their code.

G
