Subject: What could cause disappearing array?
Posted by robintw on Tue, 19 Jan 2010 21:54:25 GMT
View Forum Message <> Reply to Message

Hi,

I've got a very strange problem in one of my IDL programs. I have two nested FOR loops with some processing happening in the inside loop, and after a bit of processing one of my arrays seems to disappear.

I've checked this with some strategically placed help statements, when it seems to disappear it there (comes up with the right dimensions etc with the help command) at the bottom of the loop, but by the time the loop starts again it is showing as undefined.

I was about to tell you how many iterations of my loop this occurred after (by looking at the value of i and j when it crashed) but after it's crashed I find i and j to both be equal to 0 - even though it's gone through most of the loop already.

I can't see anywhere in the code that I'm assigning anything to i or j. I've left the FOR loop to look after them itself, yet somehow they end up as 0.

When I comment out all of the code inside the for loop then the loop runs perfectly to the end, and i and j don't get reset to zero part way through.

Does anyone have any ideas what on earth might be causing this? I've really got no idea!

The code for this function is below. If you need any of my code from other functions called by this function then just let me know.

Cheers,

Robin

```
PRO TRY_PLACING_REGION, WholeImage, edge_map, window_size
  COMMON SegData, seg_id, segment_image

  d = (window_size - 1) / 2

  image_dims = SIZE(WholeImage, /DIMENSIONS)

  ns = image_dims[0]
  nl = image_dims[1]
```

```
  FOR i = 0L, nl - 1 DO BEGIN
    FOR j = 0L, ns - 1 DO BEGIN
      print, i
      print, j
      help, segment_image

      j_lower_limit = j-d
      IF j_lower_limit LT 0 THEN j_lower_limit = 0

      j_upper_limit = j+d
      IF j_upper_limit GT ns - 1 THEN j_upper_limit = ns - 1

      i_lower_limit = i-d
      IF i_lower_limit LT 0 THEN i_lower_limit = 0

      i_upper_limit = i+d
      IF i_upper_limit GT nl - 1 THEN i_upper_limit = nl - 1

      image_moving_window = WholeImage[j_lower_limit:j_upper_limit,
i_lower_limit:i_upper_limit, *]
      edge_moving_window = edge_map[j_lower_limit:j_upper_limit,
i_lower_limit:i_upper_limit]
      segment_moving_window =
segment_image[j_lower_limit:j_upper_limit, i_lower_limit:i_upper_limit]

      ; Go to next possibility if moving window contains edges
      if total(edge_moving_window) NE 0 THEN CONTINUE

      ; If any of the window's pixels are already in a segment then go
to next possibility
      if total(segment_moving_window) NE 0 THEN CONTINUE

      inequality_result = INEQUALITY_FUNCTION(image_moving_window, [1,
1, 1])

      ; If one of the pixels had a inequality value greater than 1 then
continue
      if inequality_result EQ 0 THEN CONTINUE

      ; If we've got to here then it's a suitable seed
      print, "Found a suitable seed"

      ; Mark the seed as a segment
      segment_image[j_lower_limit:j_upper_limit,
i_lower_limit:i_upper_limit] = seg_id

      help, segment_image
```

```
      seg_id++
    ENDFOR
  ENDFOR

  ENVI_ENTER_DATA, segment_image
END
```

Subject: Re: What could cause disappearing array?
Posted by robintw on Thu, 21 Jan 2010 17:33:20 GMT
View Forum Message <> Reply to Message

On 21/01/2010 17:17, Ed Hyer wrote:
>>> "R.G. Stockwell"<noem...@please.com> wrote in message
>>>> One note: that array is in a common block, who knows who else is
>>>> manipulating it.
>
> Change the loop variables, see what happens.

Thanks for the idea. I've renamed all of the loop variables I'm using so they have descriptive names (such as row, column) rather than just i, j, k etc. However, I still get the array disappearing!

Anyone got any other ideas?

Robin

Subject: Re: What could cause disappearing array?
Posted by wlandsman on Thu, 21 Jan 2010 18:15:48 GMT
View Forum Message <> Reply to Message

On Jan 21, 12:33 pm, Robin Wilson <r.t.wil...@rmplc.co.uk> wrote:
> On 21/01/2010 17:17, Ed Hyer wrote:
>
>
> Thanks for the idea. I've renamed all of the loop variables I'm using so
> they have descriptive names (such as row, column) rather than just i, j,
> k etc. However, I still get the array disappearing!
>
> Anyone got any other ideas?

A longshot but the usual reason for a disappearing array is that you are not where you think you are, i.e. the program is crashing inside a different procedure/function (probably with similar variable names). After you crash, does IDL> help,/traceback show you to be where you think you are?

Subject: Re: What could cause disappearing array?
Posted by robintw on Thu, 21 Jan 2010 20:54:56 GMT
View Forum Message <> Reply to Message

> After you crash, does IDL> help,/traceback show you to be
> where you think you are?

Hi,

Thanks for the idea. Sadly, when I call help, /traceback it shows me to be exactly where I expect it to be.

Cheers,

Robin

Subject: Re: What could cause disappearing array?
Posted by Karl[1] on Thu, 21 Jan 2010 23:17:16 GMT
View Forum Message <> Reply to Message

On Jan 21, 4:08 pm, Mark <mark.h...@gmail.com> wrote:
> On Jan 20, 9:42 pm, Robin Wilson <r.t.wil...@rmplc.co.uk> wrote:
> ...
>
>> FOR win_size = 5, 8 DO BEGIN
>> TRY_PLACING_REGION, WholeImage, edge_map, 8
>> ENDFOR
>
> I'm curious about that code. As far as I can see, the loop index,
> win_size, is not used inside the loop, so this code has the effect of
> calling TRY_PLACING_REGION four times with the same arguments. Is that
> what you intended?

Good point, but not relevant.

What does ENVI_ENTER_DATA do with your segment_image parm?

Try checking the status of segment_image before and after the call to ENVI_ENTER_DATA.

If ENVI_ENTER_DATA clobbers that parm somehow, then that would explain why a single call to TRY_PLACING_REGION works, but more than one call does not.

Perhaps you want:

FOR win_size = 5, 8 DO BEGIN

```
    segment_image = lonarr(dims[2] + 1, dims[4] + 1)
    TRY_PLACING_REGION, WholeImage, edge_map, win_size
  ENDFOR
```

assuming that you want a fresh segment_image (zero filled) for each iteration.

Subject: Re: What could cause disappearing array?
Posted by robintw on Fri, 22 Jan 2010 21:00:40 GMT
View Forum Message <> Reply to Message

On 21/01/2010 23:17, Karl wrote:
>
> If ENVI_ENTER_DATA clobbers that parm somehow, then that would explain
> why a single call to TRY_PLACING_REGION works, but more than one call
> does not.

Well done Karl! That's the problem. I had managed to make it work earlier today when I did various alterations, including moving the ENVI_ENTER_DATA call, but now I know why it works.

Thank you very much,

Robin
