Subject: IDL 5.0 bug in SEARCH2D and SEARCH3D
Posted by David Foster on Tue, 24 Jun 1997 07:00:00 GMT
View Forum Message <> Reply to Message

IDL 5.0 bug in SEARCH2D and SEARCH3D:

Major changes have been made to SEARCH2D and SEARCH3D in IDL 5.0,
and I have sent a post to this newsgroup (and tech. support) about
a bug in SEARCH2D, but I thought I'd send a more comprehensive
report, and an email to David Fanning.

Name: David Foster
E-mail Address: foster@bial1.ucsd.edu
IDL version: { sparc sunos unix 5.0 Apr 28 1997 }
Platform and OS: Sun Sparc2, Solaris 2.5

Description of Problem:--

If you use a seed point that is on the edge of your 2D/3D array in SEARCH2D/SEARCH3D, you will get very unexpected results! Also, SEARCH3D is *much* slower in IDL 5.0 than it was in 4.0.1, at least on our platform with our data (see below).

Also, major changes have been made to both routines, including keywords being obsoleted, yet these changes are *not* reflected in the online help! (Both routines now call LABEL_REGION.PRO, so the bug could very well be there, though I doubt it.)

Also, if you read the header in SEARCH3D.PRO you see the following:

```
;   Modified:    Re-wrote to improve performance using "DILATE".
;                Obsoleted keywords INCREASE, DECREASE, and LPF_BAND.
;                Added IMAGE keyword.
```

Before you believe that the performance is increased, test the new SEARCH3D against the old one (IDL 4.0.1) on your data! When we use these routines (IDL 5.0 and 4.0.1 versions of SEARCH3D.PRO) to find an object in an INTARR(256,256,44), where the object comprises roughly 25% of the elements, here are the (approximate) results on a *very* slow Sun Sparc2:

```
4.0.1  :  101. seconds
5.0    :  612. seconds!!
```

Enough said.

Code Example:-- --

If you want to see the problem with SEARCH2D, run the following:

```
  window, xsize=512, ysize=256, /free
  array = bytarr(256,256)
  array(*) = 100             ; Object is rectangle in center
  array(0:100,*) = 0
  array(200:255,*) = 0
  array(*,0:100) = 0
  array(*,200:255) = 0
  tvscl, array, 0
  roi = search2d(array, 0, 0, 0b, 1b)   ; Find pixels between 0 and 1
  array(roi) = 1b
  tvlct, 255,0,0, 1          ; Mark pixels found as red
  tv, array, 1
```

Known Workarounds or Fixes:--

To avoid edge pixels you can do something like:

```
  xseed = (1 > xseed) < ((size(array))(1) - 2)
  yseed = (1 > yseed) < ((size(array))(2) - 2)
[ zseed = (1 > zseed) < ((size(array))(3) - 2) ]   ; For SEARCH3D
```

For SEARCH3D you can do something similar, though our solution has been to create a SEARCH3D_2.PRO that is IDL 4.0.1's SEARCH3D!! We don't like the "performance enhancement"!

RSI Technical Support Response:------------------------------------

Edge seed points have been verified as a bug.

Still waiting for response regarding slowness of SEARCH3D.

Dave
--

```
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~~
    David S. Foster         Univ. of California, San Diego
     Programmer/Analyst     Brain Image Analysis Laboratory
     foster@bial1.ucsd.edu  Department of Psychiatry
     (619) 622-5892         8950 Via La Jolla Drive, Suite 2200
                            La Jolla, CA  92037
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~~
```