RANSAC_orig in release doesn't work #523

Ⓧ Closed   **imanolooo** opened this issue on 12 Apr 2017 · 8 comments

imanolooo commented on 12 Apr 2017

First of all mention that I'm using your code in my research, customizing it.

Current Behavior

When I call the Detect function in release mode, it can not get any solution. The program never ex
the function. In debug it works properly.

Expected Behavior

Detect planes from a point cloud.

Your Environment

- OS & Version: OS X 10.11.4
- CC Version: Qt Creator 4.0.2 Based on Qt 5.7.0.
- Graphics card: NVIDIA GT750M

imanolooo commented on 12 Apr 2017

Debuging I've found that the function:

```
template< class Cell >
        inline bool BaseTree< Cell >::IsLeaf(const CellType &cell) const
        {
        return &(cell[0]) == NULL;
        }
```

returns true in debug and false in release. any idea?

asmaloney commented on 12 Apr 2017   Collab

Sounds like uninitialized data.

So maybe look at the stack when you hit that and trace back to find out what's uninitialized?

Aside: I don't include the RANSAC plugin in the macOS build because when I last looked it wasn't supported and had a ton of issues compiling cleanly. Have you had to make any changes to get it building properly?

imanolooo commented on 12 Apr 2017

Hi, thanks for answering.

I also thought on that, but i haven't found any uninit variable...
Debugging in qt creator is like a nightmare sometimes... I arrived to the problem commented. When it is creating the acceleration structure it doesn't subdivide the root because the isLeaf function return false when it should return true... In debug it worls properly... No sense on that...

Oh what a pitty that you don't include ransac on mac. So I am alone in the dark. Ransac doesn't support mac directly and I used your code thinking it work on mac. Not many adaptations I done to make it work, but it only does in debug mode...

I can pass you the adapted code if you want.

Best,

El dia 11 abr. 2017 21:21, "Andy Maloney" <notifications@github.com> va escriure:

Sounds like uninitialized data.

Some compilers will set pointers to NULL in debug builds but not in release. (This has to do with the fact that C sees declaration & initialization as two distinct steps.)

So maybe look at the stack when you hit that and trace back to find out what's uninitialized?

Aside: I don't include the RANSAC plugin in the macOS build because when I last looked it wasn't supported and had a ton of issues compiling cleanly. Have you had to make any changes to get it building properly?

—
You are receiving this because you authored the thread.
Reply to this email directly, view it on GitHub
<#523 (comment)>,

.

imanolooo commented on 12 Apr 2017

I forget to mention that if I print the values of the both sides of the comparisson i get 0x0 == 0. In debug it returns true in release false...

El dia 11 abr. 2017 21:33, "Imanol Muñoz Pandiella" < imanol.munoz.pandiella@gmail.com> va escriure:

...

dgirardeau commented on 12 Apr 2017   Me

Note that we use the original code from Ruwen Schnabel (the author of RansacSD). We haven't modified it.

It's also interesting that on Windows (Visual Studio) at least, we don't have this issue while the con does init pointers to 0 in Debug mode and left them uninitialized in Release mode...

imanolooo commented on 12 Apr 2017 • edited ▾

Yes it's quite weird, but luckily I've found a solution. I use the keyword volatile and it works. I guess compiler was affecting it with optimizations or maybe some thread... how knows!

```
template< class Cell > inline bool BaseTree< Cell >::IsLeaf(const CellType &cell) const { volati
size_t st = (size_t)(&(cell[0])); return st == 0; }
```

dgirardeau commented on 6 May 2017 • edited ▾   Me

Would the following code work? (to avoid the cast)

```
volatile const CellType* st = (&(cell[0])); return st == nullptr;
```

🏷 **dgirardeau** added the `MacOS` label on 4 Aug 2017

asmaloney commented on 3 Mar 2018   Collab

Looks like this was solved? Since RANSAC is not included with CC on macOS I'm not going to mak changes to the code.