

[博客园](#) [首页](#) [新随笔](#) [联系](#) [订阅](#) [管理](#)

一起做RGB-D SLAM (1)

• 前言

2016.11 更新

- 增加了对16.04的支持。
- 整理了过时的代码。

SLAM, 即Simultaneous Localization and Mapping, 中文译作同时定位与地图创建, 是近几十年里机器人领域有重大发展的研究方向. 作为自主机器人的核心技术, SLAM在机器人导航、控制、生产等方面都有着重要的研究意义. 尤其在二十一世纪, 以视觉传感器为中心的视觉SLAM技术, 在理论和方法上都经历了明显的转变与突破, 正逐步从实验室研究迈向成熟的市场应用. 在国内外研究如火如荼之际, 它在国内的研究尚处于起步阶段. 有关SLAM的中文资料、书籍更是难以一见. 然而, 随着机器人技术得到国家的重视, 越来越多的青年研究者、学生正逐渐踏入这片领域. 本文档则试图为这些刚走进SLAM的同事们, 提供一些简单而实际的参考.

小萝卜: 师兄! 你上面写的都是些什么东西啊!

师兄: 都是些没什么用的废话啊.....但是没这些东西文档就不上档次啊.

小萝卜: 师兄! 你别干这些无聊的事情了! 赶紧教我做SLAM啊!

师兄: 前辈才写了一段, 读者会觉得我在敷衍他们的吧. 算了, 不管了, 前两天让你跑的rgbd-slam怎么样了?

小萝卜: 跑起来了! 然后呢?

师兄: 然后你就可以调调参数, 改改代码啥的啊.

小萝卜: 师兄! 我看不懂!

师兄: 呃这个.....

小萝卜: 师兄! 你给我写一个SLAM程序吧!

师兄: 呃这个.....

小萝卜: 赶紧写啦师兄! 写完了你请我吃饭!

师兄: 吃饭啊, 那好吧.....

于是, 师兄就开始写这本文档了. 由于师兄也不知道什么时候会写完, 所以 he 每写一段就发给小萝卜看(然后请他吃饭). 还好小萝卜热情很高, 每次师兄给他写好的代码, 都拿回去仔细看而且跑了. 这也给了师兄很大动力继续往下写.

• 本文的目标

本文得带领读者自己动手去写一个RGB-D SLAM程序.

因为我相信只有通过自己写程序, 才能真正懂得slam的原理.

小萝卜: 为什么要写一个SLAM程序, 而不采用现有的代码呢?

- 师兄: 为了更深的理解.

1. 一个完整的程序含有大量的算法与GUI的代码, 你读一遍需要多久? 弄清楚原理要多久?
2. 别人工具都做好了, 代码都写完了, 参数也调好了, 你拿过来运行, 那得多给别人做一下评测.
3. 你迟早要自己写代码.

另一方面, 自己写程序, 不代表要用C++实现矩阵的线性代数. 基本的库我们还是会用的.

我们要用的库: OpenCV, PCL, g2o.

还等什么? 赶紧动手吧!

• 预备知识与编程环境

编程环境: ubuntu 12.04. 建议读者和我们使用一样或类似的环境. 如果你(出于酷或者其他什么原因)就是要用Arch/Fedora/Mac, 请你自己配环境.

必备软件的安装:

1. OpenCV

推荐从源代码安装的模式. 编译过程需要一点时间.

Step 1.

下载OpenCV源代码: <http://opencv.org/downloads.html>.

目前(2015.6)较好的版本是2.4系列, 因为3.0系列还不完善(主要是说文档). 请把它下载到电脑上随意一个目录下.

在下载过程中, 你可以安装依赖项. 基本的依赖项是那些, 直接拷贝到终端执行.

```
1 sudo apt-get install build-essential libgtk2.0-dev libjpeg-dev libtiff4-dev libjasper-dev libopenexr-dev cmake
python-dev python-numpy python-tk libtbb-dev libeigen2-dev yasm libfaac-dev libopencore-amrnb-dev libopencore-amr
wb-dev libtheora-dev libvorbis-dev libxvidcore-dev libx264-dev libqt4-dev libqt4-opengl-dev sphinx-common texlive
-latex-extra libv4l-dev libdc1394-22-dev libavcodec-dev libavformat-dev libswscale-dev
```

Step 2.

把OpenCV解压到下载目录中, 用cmake编译再安装.

```
mkdir build
cd build
cmake ..
make
sudo make install
```

编译过程需要一点时间, 长短视你机器的配置而定. 慢一点的可能一下午就过去了, 请顺便找点其他事干干例如看电影之类的.

小萝卜: 装完之后OpenCV在哪里呢?

师兄: 头文件在/usr/local/include/, 里面有opencv和opencv2的头文件. 我们基本只用opencv2嘛.

师兄: 库文件就在/usr/local/lib/下面, 当然这些在install的时候都是可以改动的. 我列的是默认位置.

小萝卜: 师兄! 刚才用的cmake是什么东西啊?

师兄: cmake就是linux下的C++管理工具啦. 简单的代码你可以用g++一条一条敲, 再多些可以用Makefile来管理. cmake就是自动生成makefile的工具. 比makefile集成度更高一些.

小萝卜: 哦好的! 我懂了师兄! 请我吃饭啦!

师兄: 好!

更新: 你也可以使用 `sudo apt-get install libopencv-dev` 安装OpenCV 2.4.9 .

2. PCL

PCL就是Point Cloud Library啦. 处理点云的必备工具.

小萝卜: 师兄! 为什么要处理点云?

师兄: 啊.....忘了说了. 这篇文档是讲RGB-D SLAM的啊. 深度相机采出来的本来就是点云数据啦.

小萝卜: 这么重要的事情为什么不放到开头去讲啊!

师兄: 我忘了.....

不管如何, PCL官网(<http://pointclouds.org>)上已经给出了ubuntu的安装方法. 因为很多开发工具在ubuntu上装起来最方便, 也比较适合小萝卜这种新手, 所以我们才选用了ubuntu.

在Ubuntu 1404或更低版本, 你需要通过私有源来安装PCL库

Step 1.

```
1 sudo add-apt-repository ppa:v-launchpad-jochen-sprickerhof-de/pcl
2 sudo apt-get update
3 sudo apt-get install libpcl-all
```

在 Ubuntu 1604, PCL库已经存在于公共软件源中, 直接安装即可:

```
sudo apt-get install libpcl-dev pcl-tools
```

但是, 在1604中PCL的配置似乎有bug, 需要在cmakelist中加一条指令修复.

师兄: 怎么样? 是不是很简单?

小萝卜: 是!

师兄: 那么, 类似的, 你能否找到PCL的头文件以及库文件的安装目录呢? 我们把这个作为留给读者的作业吧!

小萝卜: 师兄你不怕读者把你电话打爆吗?

公告

昵称: 半闲居士
园龄: 8年10个月
粉丝: 3062
关注: 0
+加关注

< 2023年1月 >											
日	一	二	三	四	五	六					
1	2	3	4	5	6	7					
8	9	10	11	12	13	14					
15	16	17	18	19	20	21					
22	23	24	25	26	27	28					
29	30	31	1	2	3	4					
5	6	7	8	9	10	11					

搜索

<input type="text"/>	<input type="button" value="找找看"/>
<input type="text"/>	<input type="button" value="谷歌搜索"/>

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

视觉SLAM(17)
机器人(14)
SLAM(13)
一起做RGB-D SLAM(7)
Kinect(4)
rgbdslam(3)
计算机视觉(2)
图像处理(2)
视觉SLAM漫谈(2)
李群(2)
更多

随机分类

随笔(2)
一起做rgbd slam(2)

随笔档案

2016年8月(1)
2016年7月(1)
2016年6月(2)
2016年3月(2)
2016年2月(2)
2016年1月(8)
2015年12月(1)
2015年8月(4)
2015年7月(4)
2015年4月(2)
2014年6月(1)
2014年4月(1)

阅读排行榜

1. 视觉SLAM漫谈(211589)
2. 一起做RGB-D SLAM (1)(137824)
3. 一起做RGB-D SLAM (2)(118148)
4. 深入理解图优化与g2o: g2o篇(115570)
5. 视觉SLAM实战(一): RGB-D SLAM V2(110972)

评论排行榜

1. 一起做RGB-D SLAM (2)(77)
2. 一起做RGB-D SLAM (5)(66)
3. 一起做RGB-D SLAM (3)(66)
4. 一起做RGB-D SLAM (6)(64)
5. 一起做RGB-D SLAM (4)(62)

推荐排行榜

1. 视觉SLAM漫谈(71)
2. 一起做RGB-D SLAM (2)(34)
3. 一起做RGB-D SLAM (1)(31)
4. 深入理解图优化与g2o: g2o篇(25)
5. 视觉SLAM漫谈(三): 研究点介绍(25)

最新评论

1. Re: 一起做RGB-D SLAM (5)

@going_go 你好 我现在也是碰到这个问题. 想问下次能怎么解决的...

--养朱的锅

2. Re: 一起做RGB-D SLAM (4)

大家好, 我在执行bin/detectFeatures时出现以下错误了oodboy@goodboy-virtual-machine: ~/slam\$ bin/detectFeatures bin/dete...

--啦啦啦啦

3. Re: 一起做RGB-D SLAM (1)

博主, 您好. 我在执行第三部分bin/detectFeatures时出现bin/detectFeatures: symbol lookup error: bin/detectFeatures: unde...

--啦啦啦啦

4. Re: 视觉SLAM实战(二): ORB-SLAM2 with Kinect2

大佬你好, 看了您的文章, 非常钦佩您VSLAM实战教程这方面的讲解. 有兴趣合作成为我们古月居网站的

到这里，必备的库就安装完了。之后若需要装别的软件，我们会单独讲。

• HELLO SLAM!

我们已经安装好了OpenCV和PCL，下面我们开始来写第一个程序吧！

小萝卜：终于可以开始写程序喽！我最爱写程序！我感到程序员之魂在我体内燃烧！

师兄：呃，可是我们第一个程序要做什么呢？

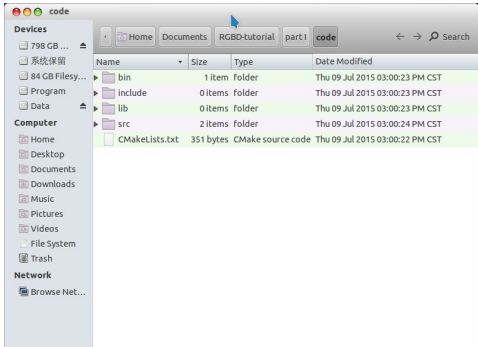
小萝卜：我们马上来写SLAM吧！

师兄：可是那样读者能看懂吗.....我们还是从简单的东西开始吧。

小萝卜：好！那就写一个简单的SLAM！

师兄：不不不，首先我们来构建一个CMake项目，作为今后写代码的模板。开头读者可能会觉得困难，但是万事开头难，后来你就慢慢习惯了。

Linux下的CMake项目通常由几个文件夹组成。例如我们今天讲的slam，你可以在机器上建一个叫slam的文件夹(注意：这个文件夹就是你代码的根目录了！)，然后在里面建几个子文件夹：



bin 用来放编译好的可执行二进制文件。

src 用来放源代码。

lib 用来放编译好的库文件。

include 用来放头文件。

为什么要用这种目录结构呢？其实这还是一种编译习惯，当然你可以把所有文件都扔一个目录里，但是这样看起来很不那么么，通常我们把源代码和编译好的东西分开，如果源代码比较多，还会按模板分开，像opencv和pcl的代码就是由好多多个模块组成的。

我们要把目录结构告诉cmake，所以我们在代码根目录下写一个CMakeLists.txt，cmake在生成代码时，会读这个文件，并按照它来编译你的代码。刚才我们对opencv进行编译时，也采用了这个步骤。好，现在在代码根目录下新建一个CMakeLists.txt：

```
touch CMakeLists.txt

写入：

CMAKE_MINIMUM_REQUIRED( VERSION 2.8 ) #设定版本
PROJECT( slam ) #设定工程名
SET( CMAKE_CXX_COMPILER "g++" ) #设定编译器

#设定可执行二进制文件的目录
SET( EXECUTABLE_OUTPUT_PATH ${PROJECT_SOURCE_DIR}/bin)

#设定存放编译出来的库文件的目录
SET( LIBRARY_OUTPUT_PATH ${PROJECT_SOURCE_DIR}/lib)
#并且把该目录设为连接目录
LINK_DIRECTORIES( ${PROJECT_SOURCE_DIR}/lib)

#设定头文件目录
INCLUDE_DIRECTORIES( ${PROJECT_SOURCE_DIR}/include)

#增加子文件夹，也就是进入源代码文件夹继续构建
ADD_SUBDIRECTORY( ${PROJECT_SOURCE_DIR}/src)
```

井号后面的是我的注释，只是为了帮助你理解，你可以不敲。通过这个文件，你应该了解了CMakeLists.txt的一些基本的用法。如果你想找一个系统的介绍，我们提供了《CMake实践》电子书，我认为是一个不错的参考资料。

小萝卜：等一下师兄！库文件和二进制都是什么啊！

师兄：二进制就是可以直接运行的程序啦，库文件呢，就是为这些二进制提供函数的啦。有main函数的代码可以编译成二进制，其他的则编译成库文件。链接时，把库文件链接到二进制上，就可以运行啦。

小萝卜：师兄我还是不懂！

师兄：呃，那我们还是通过实例来做吧。在src/文件夹下新建一个main.cpp文件，输入：

```
#include <iostream>

int main(int argc, char**argv)
{
    std::cout<<"Hello SLAM!"<<std::endl;
    return 0;
}
```

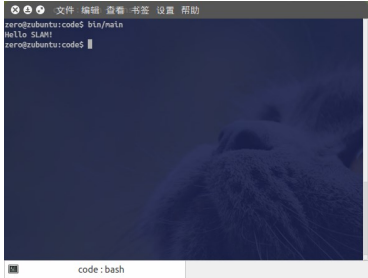
这当然是个很简单的，一目了然的程序，所以我也没有加注释。然后，我们要从这个源代码生成一个二进制。在src/目录下新建一个CMakeLists.txt，输入：

```
# 增加一个可执行的二进制
ADD_EXECUTABLE( main main.cpp )
```

这样，cmake就会把这个main.cpp编译成一个叫做main的二进制了。赶紧来试试吧。首先转到代码根目录下，输入：

```
mkdir build
cd build
cmake ..
make
```

编译通过的话，就会在bin/目录下生成一个main的二进制啦！如果你运行这个二进制，就会输出Hello SLAM的语句啦！



是不是觉得这个程序太简单了？没关系，难的在后面呢！

小萝卜：难道这一讲就这样结束了？

师兄：是啊.....毕竟写得还是蛮辛苦的，我得去休息一会。

小萝卜：可是我们基本上没什么程序啊！

师兄：别急啊，你先把cmake好好学学，之后的工作都得在这上面做。

讲师吗。官网，了解更多可以添加微信GYH-xiaogu咨询。

5. Re:一起做RGB-D SLAM (3)

请问为什么我的旋转矩阵和您的差一个负号呀

--古月居

--Mai_Qi

小萝卜: 好吧！那师兄我们下一讲要做什么呢？
师兄: 嗯，我们会用opencv读个图，再把它转成点云，怎么样？
小萝卜: 听起来不难啊，那下一讲再见啦！

本文的pdf版本、源代码、图片等资源可在我的git里下载。
网址: <https://github.com/gaoxiang12/rgb-d-slam-tutorial-gx>

标签: [SLAM](#), [机器人](#), [Kinect](#), [视觉SLAM](#)

[好文置顶](#)[关注我](#)[收藏该文](#)



半闲居士

粉丝 - 3062 关注 - 0

[+加关注](#)

31


推荐

0

反对

« 上一篇: [视觉SLAM实践\(一\): RGB-D SLAM V2](#)
» 下一篇: [一起做RGB-D SLAM \(2\)](#)

posted @ 2015-07-09 15:54 半闲居士 阅读(137826) 评论(16) 编辑 收藏 举报
[刷新评论](#) [刷新页面](#) [返回顶部](#)

 登录后才能查看或发表评论, 立即 [登录](#) 或者 [注册](#) 博客园首页

编辑推荐:

- [深入理解 Linux 物理内存分配全链路实现](#)
- [巧用视觉障眼法，还原 3D 文字特效](#)
- [MassTransit | 基于 StateMachine 实现 Saga 编排式分布式事务](#)
- [一次 SQL 调优，聊一聊 SQLSERVER 数据页](#)
- [终于弄明白了 RocketMQ 的存储模型](#)

阅读排行:

- [巧用视觉障眼法，还原 3D 文字特效](#)
- [火热的低代码到底是什么？](#)
- [C# 开发的超级屏幕类库 - 开源研究系列文章](#)
- [SQLSERVER 居然也能调 C# 代码？](#)
- [MongoDB从入门到实战之 .NET Core使用MongoDB开发ToDoList系统\(2\)-Sw](#)